SATNAC 2025

Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2025



30 November - 2 December 2025
Arabella, Hermanus, Western Cape, South Africa

2025 Proceedings

Hosted by Telkom



Africa's Ascent: Towards a Sustainable and Resilient Future enabled by secure Next-Generation Infrastructure, Digital Technology and Artificial Intelligence

Venue: Arabella, Hermanus, Western Cape, South Africa
Date: 30 November - 2 December 2025

Publication Information

Title:	Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2025
Proceedings format:	Printed
ISBN:	978-1-0492-3850-0
Editors:	Josiah Chavula and Anton Taute
Date of print:	December 2025
Version:	Final

Copyright ©2025 SATNAC. All rights reserved. This publication may be distributed, transmitted and archived for non-commercial uses in its original form without any alterations.

Responsibility for the content of the articles lies with the original authors.

SATNAC is the flagship of the Telkom Centre of Excellence (CoE) Programme.

Hosted by Telkom

Contents

	Page
SATNAC 2025 Conference Sponsors	vii
SATNAC 2025 Review Process	ix
Local Organising Committee	xi
Project Team	xi
Technical Programme Committee	xi
Index	xv
Full Papers	1

SATNAC 2025 Conference Sponsors

The SATNAC 2025 committee would like to recognise the following sponsors:

Prestige Sponsors

Executive Sponsors

Elite Sponsors

Ignite Sponsors

SATNAC 2025 Review Process

A formal Call for Papers was issued in July 2025 via the conference website and through mailing lists, inviting submissions in categories defined by the Organising Committee. Authors uploaded their papers via EasyChair, which stored submissions in a centralised database. Each paper was assigned to subject-matter experts on the review panel, who assessed suitability based on originality, relevance to the conference themes, technical or scientific merit, and clarity of presentation.

The review process adhered to the international standard for double blind peer reviews. Each submission was evaluated by a minimum of three experienced reviewers, comprising primarily esteemed South African academics and several international experts. Reviewers provided detailed feedback, both constructive and critical, to help authors refine their work. Authors received this feedback as the sole disclosure from the review process, while all other information remained confidential.

Reviewers submitted their scores and comments through the EasyChair web interface, where the Technical Programme Committee (TPC) compiled reports and calculated weighted averages for each paper. Based on these scores and the conference program's capacity, papers were ranked and selected for inclusion. Out of 153 full-paper submissions, 52 were accepted, reflecting an acceptance rate of about 34%.

Authors of accepted papers were asked to submit revised versions addressing the reviewers' feedback. Only submissions deemed to meet the high-quality standards, as recommended by the reviewers, were included as full reviewed papers in the SATNAC 2025 proceedings.

Additionally, the conference welcomed two-page work-in-progress papers, which underwent a less rigorous review process. These submissions were accepted as posters and presented at the conference but were not included in the official conference proceedings.



Dr Josiah Chavula
Chairperson
Technical Programme Committee
SATNAC 2025

Local Organising Committee

Dr Mmaki Jantjies (Chairperson)
Mr Lebo Masalesa
Ms Charlene Abrey

Project Team

Dr Maletsabisa (Tsabi) Molapo
Ms Christobelle Loock
Mr Mooketsi Mocumi
Dr Sunil Piyarlall
Dr Babedi Modieginyane
Ms Asavela Sigonya
Mr Johan Fick
Ms Moipone Tsaoane
Mr Peter Coleman

Technical Programme Committee

Dr Josiah Chavula (TPC Chairperson), University of Cape Town
Mr Anton Taute (TPC Vice Chairperson), Openserve

Mr Emmanuel Adigun, ABSA
Dr Olusola Olajide Ajayi, Adekunle Ajasin University
Dr Rodolfo Martinez-Manuel, Centro de Investigaciones en Óptica A.C.
Dr Albert Lysko, Council of Scientific & Industrial Research
Prof Bakhe Nleya, Durban University of Technology
Prof Thomas Magedanz, Fraunhofer FOKUS Institute
Prof Brenda Scholtz, Nelson Mandela University
Dr David Waswa, Nelson Mandela University
Dr Dieter Vogts, Nelson Mandela University
Dr Ifeoluwapo Fashoro, Nelson Mandela University
Prof Janet Wesson, Nelson Mandela University
Mr Lester Cowley, Nelson Mandela University
Dr Lynette Barnard, Nelson Mandela University
Dr Simone Beets, Nelson Mandela University
Prof Barry Irwin, Noroff University College
Prof Albert Helberg, North-West University
Dr Annette van der Merwe, North-West University
Mr Herman Blackie, North-West University
Dr Melvin Ferreira, North-West University
Prof Tiny Du Toit, North-West University
Mr Clarence Mthukwane, Openserve
Mr Sandile Dlamini, Openserve
Mr Tshiamo Tsabone, Openserve
Mr Taahir Patel, Openserve
Mr Yashil Morar, Openserve
Dr James Whitehead, Reutech Communications

Dr Dane Brown, Rhodes University
Prof George Wells, Rhodes University
Prof Ingrid Siebörger, Rhodes University
Mr James Connan, Rhodes University
Mr Joshua van Staden, Rhodes University
Prof Karen Bradshaw, Rhodes University
Mr Marc Marais, Rhodes University
Prof Nomusa Dlodlo, Rhodes University
Dr Stones Dalitso Chindipha, Rhodes University
Dr Yusuf Motara, Rhodes University
Dr Zelalem Shibeshi, Rhodes University
Dr Emmanuel Migabo, Tshwane University of Technology
Mr Gustave Udahemuka, Tshwane University of Technology
Dr Prosper Sotenga, Tshwane University of Technology
Prof Thomas Olwal, Tshwane University of Technology
Dr Sajid Sheikh, University of Botswana
Dr Bessie Malila, University of Cape Town
Dr Daniel Ramotsoela, University of Cape Town
Prof Hussein Suleman, University of Cape Town
Dr Joyce Mwangama, University of Cape Town
Mr Martin Thodi, University of Cape Town
Dr Caroline Gurajena, University of Fort Hare
Mr Mphenduli Ntsizi, University of Fort Hare
Prof Nobert Jere, University of Fort Hare
Mr Phumzile Nomnga, University of Fort Hare
Mr Sikhumbuzo Ngwenya, University of Fort Hare
Mr Siyabulela Dyakalashe, University of Fort Hare
Mr Michael Grobler, University of Johannesburg
Dr Ayodeji James Bamisaye, University of KwaZulu-Natal
Mr Bashan Naidoo, University of KwaZulu-Natal
Mr Bhekisizwe Mthethwa, University of KwaZulu-Natal
Prof Hongjun Xu, University of KwaZulu-Natal
Mr Jonathan Parker, University of KwaZulu-Natal
Prof Jules Tapamo, University of KwaZulu-Natal
Prof Narushan Pillay, University of KwaZulu-Natal
Prof Pradeep Kumar, University of KwaZulu-Natal
Mr Ray Khuboni, University of KwaZulu-Natal
Dr Seare Rezenom, University of KwaZulu-Natal
Dr Sulaiman Patel, University of KwaZulu-Natal
Prof Tahmid Quazi, University of KwaZulu-Natal
Mr Olumide Alamu, University of Lagos
Mr Jonas Manamela, University of Limpopo
Prof Mthulisi Velempini, University of Limpopo
Mr Promise Malatji, University of Limpopo
Mr Simon Ramalepe, University of Limpopo
Dr Thipe Modipa, University of Limpopo
Mr Tumisho Mokgonyane, University of Limpopo
Dr Allan de Freitas, University of Pretoria
Prof Herman Myburgh, University of Pretoria
Mr Kealeboga Mokise, University of Pretoria

Prof Masike Malatji, University of South Africa
Dr Mbuyu Sumbwanyambe, University of South Africa
Dr Riaan Wolhuter, University of Stellenbosch
Prof Thomas Niesler, University of Stellenbosch
Mr Willem Smit, University of Stellenbosch
Mr André Henney, University of the Western Cape
Dr Charley Lewis, University of the Western Cape
Prof Louise Leenen, University of The Western Cape
Prof Mehrdad Ghaziasgar, University of The Western Cape
Mr Mfundo Maneli, University of The Western Cape
Dr Olasupo Ajayi, University of The Western Cape
Dr Oluwakorede Oluyide, University of The Western Cape
Prof Omowunmi Isafiade, University of The Western Cape
Dr Chabalala Chabalala, University of the Witwatersrand
Prof Fambirai Takawira, University of the Witwatersrand
Prof Jules Moualeu, University of the Witwatersrand
Prof Ling Cheng, University of the Witwatersrand
Dr Mitchell Cox, University of the Witwatersrand
Prof Olutayo Oyerinde, University of the Witwatersrand
Prof Alfredo Terzoli, University of Zululand
Ms Ijeoma Ezeji, University of Zululand
Prof Matthew Adigun, University of Zululand
Ms Nombuso Sibeko, University of Zululand
Mr Paul Tarwireyi, University of Zululand
Dr Pragasen Mudali, University of Zululand
Dr André Joubert, Vaal University of Technology
Dr Temidayo Otunniyi, Vaal University of Technology

SATNAC 2025 Technical Programme

Artificial Intelligence and Data-Driven Applications

Title:	**1089. Quasi-Distributed Fiber Bragg Grating Sensor Networks for First-Principles Impact Localization in Bridge Infrastructure**	Page 2
Authors:	Natinale Yosef, Michael F Grobler, Ishana Ranchod	
Title:	**1199. IsiNdebele Question-Answering Model using Minimal Dataset**	Page 12
Authors:	Promise Malatji, Thipe Modipa	
Title:	**1332. Comparing Video Classification Networks in Isolated Sign Language Recognition**	Page 20
Authors:	Luke Goodall, James Connan, Dane Brown, Marc Marais	
Title:	**1938. Investigating the Effect of Copy-paste Augmentation on Fish Detection Model Generalisation**	Page 29
Authors:	Kenneth Chieza, Dane Brown, Daanyaal Salie, James Connan	
Title:	**2212. Accelerating Marine Ecosystem Analysis: A Diversity-Driven Active Learning Pipeline for Fish Detection**	Page 36
Authors:	Daanyaal Salie, Dane Brown, James Connan, Kenneth Chieza	
Title:	**4781. Lang2Vec: language-level embeddings for robust low-resource language identification in text**	Page 45
Authors:	Erwin Schillack, Thomas Niesler	
Title:	**4974. Textureless-Resilient Propagation Matching in Multi-View Stereo (TPM-MVS)**	Page 53
Authors:	Ray Khuboni, Hongjun Xu	
Title:	**5062. Investigating Adaptive Fine-Tuning Strategies for Isolated Sign Language Recognition**	Page 63
Authors:	Marc Marais, Dane Brown, James Connan	
Title:	**6284. Knowledge Distillation Compression of a State-of-the-Art Saliency Object Detection Model**	Page 71
Authors:	Robin Hayman, Allan De Freitas, Herman Myburgh	
Title:	**6708. Evaluating Model Compression Techniques for Efficient Edge AI Deployment**	Page 82
Authors:	Mondli Zungu, Skhumbuzo Zwane, Matthew Adigun	
Title:	**7370. Improving Sleep: An Analysis of AI-driven Sleep Monitoring Capabilities in Consumer-grade Smartwatches**	Page 88
Authors:	Tadiwanashe Musemesi, Brenda Scholtz, Simone Beets	
Title:	**7909. MobileNetV3-Based Disease Detection from Poultry Faecal Images Using a Mobile-Integrated System**	Page 97
Authors:	Tinashe Crispen Garidzira, Khulumani Sibanda, Tamuka Nyashadzashe	
Title:	**8024. WhaleVAD-BPN: Improving Baleen Whale Call Detection with Boundary Proposal Networks and Post-processing Optimisation**	Page 105
Authors:	Christiaan Geldenhuys, Günther Tonitz, Thomas Niesler	
Title:	**8188. Traffic Sign Recognition: Comparing Custom CNNs and Transfer Learning Approaches for Autonomous Vehicle Systems**	Page 115
Authors:	Roland Chuma, Joshua van Staden, Dane Brown	
Title:	**9645. CACOphony: Contrastive Architecture for TB Cough classification**	Page 123
Authors:	Minette Farrell, Grant Theron,Thomas Niesler	

Core Network Technologies and Architecture

Title:	**1317. Deep Learning vs Traditional Learning for Ransomware Detection: A Performance Comparison Analysis**	Page 130
Authors:	Emmanuel Musiiwa, Stones D Chindhipha, Moses Moyo	
Title:	**2263. Comparison of robustness of modulation classifiers against white-box adversarial attacks**	Page 140
Authors:	Arnold van der Merwe, Albert Helberg	
Title:	**2790. Experimental Characterisation of an Erbium-Doped Fibre Amplified Spontaneous Emission Source for Sensing Systems**	Page 150
Authors:	Josue Kalolo Kalolo, Michael Federick Grobler, Jean-Jacques Monga Kaboko	
Title:	**4949. Development of an architecture to detect DDoS attacks in end-to-end Network Slicing on 5G Networks**	Page 156
Authors:	Phathuvuyo Joyi, Caroline Gurajena, Moshe Masonta	
Title:	**5317. Building a low-cost cloud native 5G network slicing experimental Testbed: Open-Source Solutions, Lessons Learned and Future Directions**	Page 164
Authors:	Humphrey Owuor, Bessie Malila, Joyce Mwangama, Lusani Mamushiane, Maurine Chepkoech, Tariro Mukute, Thomas Magedanz, Elena-Ramona Modroiu	

Data Centers, Cloud and Edge Computing

Title:	**1459. Digital Twin Implementation for Real-time Structural Health Monitoring**	Page 175
Authors:	Filmon Gebremichael, Michael Grobler, Sune Von Solms	
Title:	**2412. Digitization of Analogue Gauges using Computer Vision**	Page 185
Authors:	Erico Kayumba Mutonkole, Olasupo Ajayi, Omowunmi Isafiade	
Title:	**2701. A Chronological Analysis of the Evolution of SmartNICs**	Page 194
Authors:	Olasupo Ajayi, Ryan Grant	
Title:	**6414. Designing a Plant Leaf Disease Detection System using Image Processing and Machine Learning Techniques**	Page 202
Authors:	Luvolwethu Tokwe, Nobert Jere, Khulumani Sibanda	
Title:	**7133. Developing a Large Language Model for Text Generation in Under-Resourced Sepedi language**	Page 209
Authors:	Mathews Tladi, Jonas Manamela, Thipe Modipa	
Title:	**8582. A Cross-device Horizontal Federated Learning with an asynchronous-based aggregation mechanism in detecting Secure Shell Brute Force attacks for Internet of Things in Edge Computing**	Page 217
Authors:	Miyelani Silence Madiba, Mthulisi Velempini	

Network Operations and Management; Network Standards and Regulatory

Title:	**2712. A Comparison of IEEE 1900.5.2 Spectrum Consumption Models with Conventional Methods for Coexistence Analysis**	Page 226
Authors:	Schuyler Scholtz, Melvin Ferreira, Herman Blackie	
Title:	**6254. Best Fit Channel Switching for RF Energy Harvesting Cognitive Radio Networks**	Page 236
Authors:	Fidel Wasonga, Thomas Olwal, Anish Kurien	
Title:	**6925. Assessing the Performance of Wi-Fi Student Networks at the University of Zululand's KwaDlangezwa Campus**	Page 244
Authors:	Dumile Penelope Gumede, Alfredo Terzoli, Nombuso Cecilia Sibeko	
Title:	**6990. Multi-Objective Optimization for Trust-Based Relay Selection in 6G-IoT Sidelink Communication**	Page 251
Authors:	Stephen Ekwe, Joyce Mwangama	

Title: **9941. Estimating QoE of Encrypted Video Streams, a Machine Learning Approach** Page 262
Authors: Best Nkhumeleni, Joyce Mwangama

Security, Privacy, and Trust

Title: **0198. Investigating the Robustness of Audio Features Under Concept Drift in Android Malware Detection** Page 270
Authors: Mussa Phiri, Paul Tarwireyi, Alfredo Terzoli

Title: **1037. A Joint Federative Learning and Zero Trust-Based Privacy Enhancement Algorithm** Page 279
Authors: Beverly Mosenapelo Pule, Thoriso Nleya, Khulekani Sibiya, Bakhe Nleya

Title: **5652. Enhancing IoT Security with Anomaly Detection: An Interpretable Multilayer Perceptron Approach** Page 288
Authors: Nathan Morgenrood, Tiny Du Toit, Linda Redelinghuys

Title: **6668. A Comparative Evaluation of AI Models for Fraud Detection Using Real-World Malawian Banking Data** Page 298
Authors: Mwayi Mwalwimba, Dieter Vogts, Janet Wesson

Title: **7814. Deep Learning Driven Threat Detection in Network Traffic Using Multi-Layer Perceptron (MLP) on Public Intrusion Datasets** Page 308
Authors: Tinashe Crispen Garidzira, William Tichaona Vambe, Courage Matobobo, Caroline Gurajena

Sustainability and Social Impact

Title: **0691. Offline-First Mobile Application for Communal Livestock Management: A Participatory Design Approach in South Africa's Eastern Cape** Page 317
Authors: Oluwatobi John Aduradola, Caroline Gurajena, Dare Fatumo

Title: **1000. Fibre-Optic Interferometric Sensor System for Real-Time Structural Health Monitoring** Page 325
Authors: Sandisiwe Bangani, James Jena, David Waswa

Title: **2961. On the EVT-Based Nonlinear Impulsive Noise Suppression Over Indoor Low-Voltage Networks** Page 333
Authors: Steven O. Awino, Bakhe Nleya, Mlungisi Molefe

Title: **4019. Rethinking Graduate Skills Integration for the Fourth Industrial Revolution in South African Higher Education** Page 338
Authors: Rennie Naidoo, Naomi Isabirye, Olorunjuwon Michael Samuel

Title: **4657. Development of a Curvature Sensing System using Fibre Bragg Grating Sensors** Page 345
Authors: Odendaal Mudondo, Michael F. Grobler, Natinale Yosef

Title: **4833. Ensemble Learning and Socio-Economic Indicators for Forecasting South African School Performance: An Experimental Analysis** Page 355
Authors: Njabulo Ndlela, Sizakele Mathaba, Alfredo Terzoli

Title: **5275. Polarization-based Optical Fibre Sensor for Landslide Monitoring** Page 365
Authors: Sisipho Dlakiya, Francois Swanepoel, James Jena, Gaathier Mahed, David Waswa

Title: **6628. OFDR Fibre Optic Sensor for Sinkhole Sensing in a Simulated Environment** Page 373
Authors: Francois Swanepoel, James Jena, Gaathier Mahed, David Waswa, Victor Agbakoba

Title: **7796. A Review of Maximum Power Point Tracking Methods for Wind Energy Conversion Systems** Page 379
Authors: Floyd Mabiala, Clement Nyirenda, Mohamed Kahn

Title: **9703. Tiny-CNN: Achieving State-of-the-Art Solar Panel Soiling Detection on Edge Devices through Knowledge Distillation and Quantization** Page 389
Authors: Tyron Ncube, Nomusa Dlodlo, Josephat Kalezhi, Alfredo Terzoli

Wireless Communications and Next-Generation Technologies

Title: Authors:	**2252. Outage Performance Analysis of Green Energy IoT Network** Neo Selematsela, Chabalala Chabalala, Fambirai Takawira	Page 397
Title: Authors:	**2802. Evaluation Framework for LMS-Based Channel Estimation in MIMO-NOMA Systems** Lehlohonolo Sekokotoana, Fambirai Takawira, Olutayo Oyerinde	Page 405
Title: Authors:	**2947. Join-Aware Energy Modelling and Optimisation of LoRaWAN Nodes for Underground Bulk Water Metering** Advisor M. Dlamini, Emmanuel M. Migabo, Mbuyu M. Sumbwanyambe, Prosper Z. Sotenga	Page 413
Title: Authors:	**3229. FLASH - An Alternative Approach To Wideband Spectrum Sensing in 5G Environments** Albertus B. Jooste, Albert Helberg	Page 423
Title: Authors:	**9517. A Clustering-Based Multi-Layer Optimization Framework for Energy Efficiency and Congestion in 5G-Enabled IoT Networks** Basil Dlamini, Emmanuel Migabo, Anish Kurien	Page 433
Title: Authors:	**9847. Validation of an OpenAirInterface 5G Testbed for Uplink-Heavy Live Audio-Visual Production** Andries Oelofse, Melvin Ferreira, Herman Blackie	Page 443

Artificial Intelligence and Data-Driven Applications

Quasi-Distributed Fiber Bragg Grating Sensor Networks for First-Principles Impact Localization in Bridge Infrastructure

Natinale Yosef
Department of Electrical and Electronic Engineering Sciences
University of Johannesburg
, Johannesburg, South Africa,
202000323@student.uj.ac.za

Michael Grobler
Department of Electrical and Electronic Engineering Sciences
University of Johannesburg
Johannesburg, South Africa
michaelg@uj.ac.za

Ishana Ranchod
Department of Civil Engineering Science
University of Johannesburg
Johannesburg, South Africa
217002014@student.uj.ac.za

***Abstract*— Accurate impact localization on structures is critical for real-time structural health monitoring (SHM) and incident response systems. This paper presents a First-Principal approach for impact localization using Fiber Bragg Grating (FBG) sensors. A 4-sensor FBG network was deployed across a 10-meter bridge span with sensors positioned at 2m, 4m, 6m, and 8m. Impact tests were conducted at four positions using 80kg and 40kg, loading conditions. Three physics-based algorithms were developed and validated: Time Difference of Arrival (TDOA), Energy Decay, and a Hybrid approach combining both methods. Wave speed estimation yielded 508 m/s through least-squares regression analysis with correlation coefficient $R^2 = 0.611$. Results demonstrate loading dependent performance characteristics, with 80kg conditions achieving superior localization accuracy across all algorithms. The hybrid method achieved optimal performance with 1.35m mean absolute error (13.5% relative error) under 80kg loading, while energy decay methods maintained 100% success rates across both loading conditions. TDOA methods showed high sensitivity to signal amplitude, with success rates declining from 75% to 25% under reduced loading. The study validates that physics-based algorithms can achieve practical localization accuracy suitable for structural monitoring applications when adequate signal strength is maintained, providing a foundation for interpretable and robust impact detection systems in bridge infrastructure.**

Keywords—Fiber Bragg Grating Sensors, Impact Localization Structural Health Monitoring, Wave Propagation, First- Principal

I. INTRODUCTION

Bridge infrastructures are critical components of transportation networks worldwide, with structural integrity directly impacting public safety and economic activity [1]. In South Africa, the need for effective structural health monitoring is becoming increasingly urgent. Recent studies using Weigh-in-Motion (WIM) data reveal that many bridges, particularly the short to medium span, are subjected to load effects significantly higher than expected, in some cases nearly double the anticipated values. With heavy freight traffic dominating national routes and a growing reliance on road transport due to limitations in the rail system, the long-term reliability of bridges is under mounting pressure [2]. Traditional inspection methods rely primarily on visual assessment and periodic load testing, approaches that are labor-intensive, costly, and inherently limited in their ability to detect damage in real-time. The development of automated monitoring systems capable of detecting, locating, and characterizing structural impacts represents a significant advancement toward proactive SHM systems.

Current approaches to impact localization in civil structures employ various sensor technologies including accelerometers, strain gauges, and piezoelectric sensors, often coupled with machine learning algorithms for position estimation [3], [4], [5]. While these methods have demonstrated success in controlled laboratory environments, their practical implementation faces significant challenges. Machine learning approaches typically require large amounts of training datasets encompassing numerous impact scenarios, sensor configurations, and environmental conditions to achieve reliable performance. Additionally, the lack of transparency of many ML algorithms makes it difficult to understand how predictions are derived from input data, reducing their acceptance among practicing engineers who require interpretable results for decision-making processes.

Fiber Bragg Grating (FBG) sensors offer distinct advantages for structural monitoring applications, including distributed sensing capabilities, electromagnetic immunity, and long-term stability in harsh environments [6], [7].The propagation of elastic waves through structural elements follows well-established physics principles, suggesting that localization algorithms based on wave propagation theory may provide more robust and interpretable solutions than purely data-driven approaches. Time Difference of Arrival (TDOA) methods utilize the principle that acoustic waves reach sensors at different times based on geometric relationships, while energy decay models exploit the physical relationship between signal amplitude and distance from the source [8], [9].These first-principal approaches require minimal training data and provide direct insight into the underlying structural response mechanisms.

This paper presents a systematic investigation of first-principal impact localization algorithms for bridge structures instrumented with FBG sensor networks. Three distinct approaches are developed and validated: a TDOA method based on cross-correlation analysis, an energy decay method utilizing geometric spreading principles, and a hybrid algorithm combining both techniques. The algorithms are validated through controlled impact testing on a 10-meter instrumented bridge span with four FBG sensors positioned at 2m intervals. Wave propagation speed estimation methodology is developed through least-squares regression analysis, providing a foundation for TDOA calculations. Performance evaluation demonstrates that physics-based approaches can achieve practical localization accuracy while maintaining reliable impact detection across multiple test conditions.

The contributions of this work include: (1) development and validation of three first-principal localization algorithms suitable for sparse FBG sensor networks, (2) establishment of wave speed estimation methodology for structural applications, (3) comprehensive performance comparison revealing algorithm-specific advantages and limitations, and (4) demonstration of practical localization accuracy suitable for real-world structural monitoring applications. The results provide a foundation for physics-informed monitoring systems that combine sensor technology with fundamental structural mechanics principles.

II. BACKGROUND

A. Structural Health Monitoring Fundamentals

Structural health monitoring systems aim to provide real-time assessment of structural condition through continuous sensing and analysis of structural response parameters. The fundamental premise of SHM is that structural damage or impact events alter the system's dynamic characteristics, creating detectable changes in measured responses such as strain, displacement, or vibration patterns [10]. Impact localization represents a specific application within the broader SHM framework, focusing on determining the spatial coordinates of external loading events based on distributed sensor measurements.

The effectiveness of any impact localization system depends on three primary factors: sensor technology characteristics, signal processing methodology, and the underlying physical relationship between impact location and measured response. Traditional approaches have relied heavily on dense sensor networks to provide adequate spatial resolution, often requiring 10-20 sensors per span to achieve acceptable accuracy [11]. This requirement poses significant economic barriers to practical implementation, particularly for large-scale infrastructure monitoring applications.

B. Fiber Bragg Grating Sensor Technology

Fiber Bragg Gratings represent a mature optical sensing technology based on periodic refractive index variations inscribed in optical fiber cores. When subjected to mechanical strain or temperature changes, the Bragg wavelength shifts according to the relationship:

$$\frac{\Delta\lambda_B}{\lambda_B} = (1 - p_e)\varepsilon + (\alpha + \xi)\Delta T \quad (1)$$

Where $\Delta\lambda_B$ is the wavelength shift, λ_B is the Bragg wavelength, p_e is the photo-elastic coefficient, ε is the applied strain, α is the thermal expansion coefficient ξ is the thermo-optic coefficient, and ΔT is the temperature chang [12].

The primary advantages of FBG sensors for structural monitoring include: (1) immunity to electromagnetic interference, (2) capability for distributed sensing along single optical fibers, (3) multiplexing potential allowing dozens of sensors per fiber, (4) long-term stability in harsh environments, and (5) high sensitivity to mechanical strain. These characteristics make FBG sensors particularly suitable for permanent installation in civil infrastructure, where traditional electrical sensors may face durability challenges [13], [14], [15].

C. Wave Propagation in Structural Elements

Impact events generate elastic waves that propagate through structural elements at velocities determined by material properties and structural geometry. For flexural waves in beam-like structures, the relationship between wave speed, frequency, and material properties follows:

$$c = \sqrt[4]{\frac{4EI\omega^2}{\rho A}} \quad (2)$$

Where c is the wave speed, E is the elastic modulus, I is the second moment of area, ω is the angular frequency, ρ is the material density, and A is the cross-sectional area [16].However, for practical impact localization applications, empirical wave speed determination often provides more reliable results due to the complexity of multi-mode propagation in real structures.

The attenuation of elastic waves in structural elements occurs through geometric spreading and material damping. For cylindrical spreading in plate-like structures, the amplitude decay relationship can be approximated as:

$$A(r) = A_0 \frac{e^{-\alpha r}}{\sqrt{r}} \quad (3)$$

Where $A(r)$ is the amplitude at distance r, A_0 is the source amplitude, and α is the material attenuation coefficient [17]. This relationship forms the theoretical foundation for energy-based localization algorithms.

D. Time Difference of Arrival Methods

TDOA localization methods exploit the fundamental principle that elastic waves reach different sensor locations at times proportional to their respective distances from the impact source. For a one-dimensional localization problem with sensors at known positions, the impact location x can be determined by solving the system of equations:

$$t_i - t_j = \frac{|x - x_i| - |x - x_j|}{v} \quad (4)$$

Where t_i and t_j are arrival times at sensors i and j, x_i and x_j are sensor positions, and v is the wave propagation speed [18],[19].

The accuracy of TDOA methods depends critically on precise arrival time determination and accurate wave speed estimation. Cross-correlation analysis provides a robust approach for arrival time detection, particularly in noisy environments where simple threshold-based methods may fail. The cross-correlation function between two sensor signals is defined as:

$$R_{xy}(\tau) = \int_{-\infty}^{\infty} x(t)y(t+\tau)dt \quad (5)$$

The time lag τ that maximizes $|R_{xy}(\tau)|$ corresponds to the relative arrival time between the two sensors [19].

E. Energy-Based Localization Methods

Energy-based localization methods utilize the relationship between signal energy and distance from the impact source. Assuming isotropic wave propagation and geometric spreading losses, the energy measured at senso i can be modeled as:

$$E_i = E_0 \frac{G(f)}{r_i^n} \quad (6)$$

Where E_0 is the source energy, G(f) represents frequency-dependent propagation effects r_i is the distance from source to sensor i, and n is the geometric spreading exponent (typically 1-3 depending on wave type and structure geometry) [20].

Energy-based methods offer advantages in scenarios where precise arrival time determination is challenging due to low signal-to-noise ratios or complex multi-mode propagation. The method requires estimation of both the source energy E_0 and the spreading exponent n, which can be accomplished through nonlinear optimization techniques [20].

F. Previous Work in Impact Localization

Recent research in physics-based impact localization has focused on aerospace and composite applications, where controlled experimental conditions enable algorithm development and validation. De Simone *et al.* [21] proposed a monitoring system for aluminum and composite structures that achieved accurate real-time localization using optimized sensor configurations and Akaike Information Criterion–based time-of-arrival estimation. Similarly, Coles *et al.* [22] investigated time reversal methods in fiber-reinforced composites, introducing a minimum average approach that improved localization accuracy and robustness even under conditions such as double impacts and material damage.

However, civil infrastructure applications present unique challenges including environmental variability, structural complexity, and the need for long-term reliability with minimal maintenance. The translation of laboratory-validated algorithms to field applications requires careful consideration of these factors, particularly regarding wave speed estimation and sensor calibration procedures [23].

FBG sensors have been widely implemented for impact localization because of their small size, multiplexing capability, and immunity to electromagnetic interference. Yu and Liang [24] demonstrated that FBGs embedded on composite plates can capture low-velocity impact responses, which can then be analyzed using recurrence quantification analysis (RQA) and genetic algorithms to estimate impact positions with average localization errors of 25 mm. More recent work by Duan et al. [25] emphasized that the spatial arrangement of FBG arrays plays a critical role in localization accuracy. By optimizing sensor placement to account for the elliptical sensitivity region of FBGs, they improved localization performance from 74% to over 90% in controlled tests. Together, these studies highlight that both advanced signal-processing methods and optimized array layouts are key to achieving reliable and practical FBG-based impact localization.

III. METHODOLOGY

A. Experimental Setup

The experimental validation was conducted on a 10-meter instrumented bridge span equipped with a distributed FBG sensor network. Four FBG sensors (designated Ch01Gr01 through Ch01Gr04) were positioned at 2m intervals along the bridge deck at locations x = 2m, 4m, 6m, and 8m from the reference origin. The sensors were surface mounted to the bridge deck using strain gauge glue to ensure effective strain transfer from the structural element to the optical fiber Figure 1 presents the bridge setup with the sensor placement.

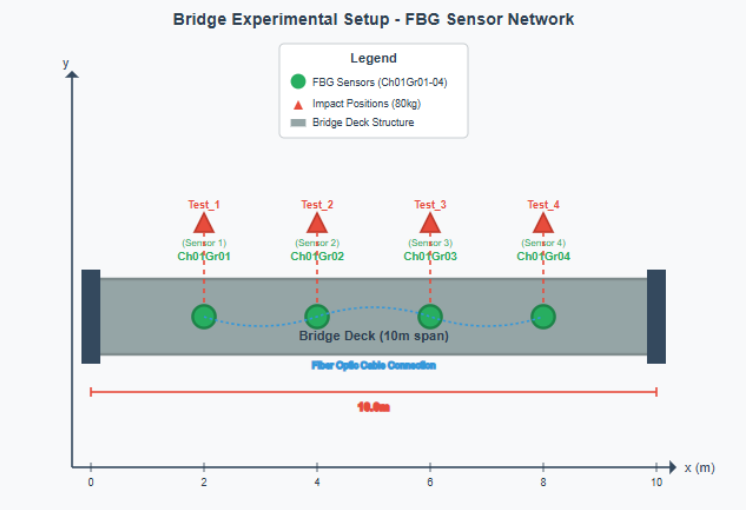


Figure 1: Bridge and Sensor Setup

Impact loading was applied using a standardized 40kg mass dropped from a consistent height to generate repeatable impact conditions. Four impact positions were tested, corresponding to locations directly above each sensor position (2m, 4m, 6m, and 8m). Each test configuration was designated as Test-1 through Test -4. The loading is then repeated with an 80kg load.

Data acquisition was performed using a high-speed FBG interrogation system (SmartScan Dynamic interrogator) where the wavelength data is sampled at 1.25 Khz. Wavelength shift measurements were converted to strain using standard FBG calibration relationships, and baseline correction was applied to remove thermal drift and static loading effects.

B. Signal Processing and Preprocessing

Raw FBG data underwent a multi-stage preprocessing pipeline to isolate impact-induced responses from background noise and environmental effects. The preprocessing sequence consisted of:

1) Baseline Establishment: A 1-second baseline period was extracted from 2-3 seconds after data acquisition initiation, well before the expected impact time of approximately 5-7 seconds. Baseline statistics (mean and

standard deviation) were calculated for each sensor channel to establish reference conditions,Figure 2 below shows an example of how the Baseline is established with one of the sensors.

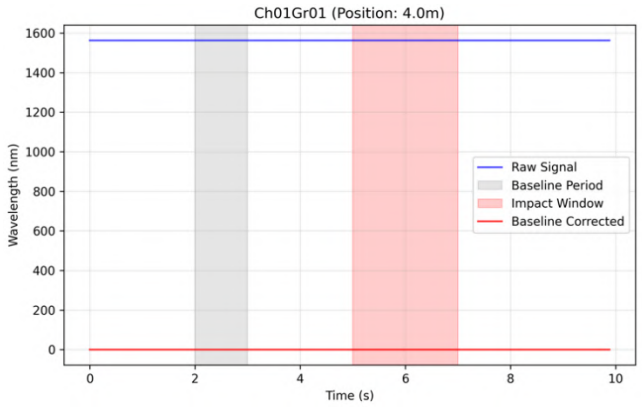


Figure 2 : Baseline Establishment on the sensors data.

2) Impact Detection**:** Impact events were identified using a threshold-based detection algorithm with the threshold set to 4σ above the baseline noise level

$$\text{Threshold}_i = \mu_{\text{baseline},i} + 4\sigma_{\text{baseline},i} \quad (7)$$

$\mu_{\text{baseline},i}$ and $\sigma_{\text{baseline},i}$ are the baseline mean and standard deviation for sensor i.

3) Signal Segmentation: Detected impact events were segmented into analysis windows extending from 2 seconds before to 3 seconds after the estimated impact time. Baseline correction was applied to all segmented signals:

$$s_{\text{corrected},i}(t) = s_{\text{raw},i}(t) - \mu_{\text{baseline},i} \quad (8)$$

4) Data Validation: Sensor channels exhibiting excessive zero values (>10% of samples) or abnormally low signal variance were excluded from analysis to ensure data quality.

C. Wave Speed Estimation

Accurate wave speed estimation represents a critical component of first-principal localization algorithms. The methodology employed a least-squares regression approach utilizing known impact-sensor distance relationships from all test configurations.

For each test, relative arrival times were calculated using cross-correlation analysis between sensor pairs. The time difference between the earliest and subsequent arrivals was computed as:

$$\Delta t_i = t_{\text{arrival},i} - \min\left(t_{\text{arrival},j}\right) \text{ for all } j \quad (9)$$

Distance differences were calculated based on known geometric relationships:

$$\Delta d_i = \left|x_{\text{impact}} - x_{\text{sensor},i}\right| - min\left(\left|x_{\text{impact}} - x_{\text{sensor},j}\right|\right) for\ all\ j \quad (10)$$

Wave speed was estimated through linear regression of the distance-time relationship:

$$v = \frac{\sum_k \Delta d_k \cdot \Delta t_k}{\sum_k \Delta t_k^2} \quad (11)$$

where the summation extends over all valid time-distance pairs from all test configurations.

D. Localization Algorithms

Three distinct physics-based algorithms were developed and implemented for impact localization analysis.

1) Time Difference of Arrival (TDOA) Method: The TDOA algorithm exploits the fundamental relationship between impact position and wave arrival times at distributed sensors. Arrival times were determined through cross-correlation analysis using the sensor exhibiting maximum energy response as the reference:

$$R_{\text{ref},i}(\tau) = \int_{-\infty}^{\infty} s_{\text{ref}}(t)\, s_i(t+\tau)\, dt \quad (12)$$

The time lag maximizing $|R_{\text{ref},i}(\tau)|$ provided the relative arrival time between the reference sensor and sensor i. Impact position was determined by solving the nonlinear optimization problem:

$$\hat{x}_{impact} = \arg\min_x \sum_{i=1}^{N} \left[\Delta t_{\text{measured},i} - \frac{\left|x - x_{\text{sensor},i}\right| - \left|x - x_{\text{ref}}\right|}{v}\right]^2 \quad (13)$$

subject to the constraint $0 \le x \le L_{bridge}$, *where* L_{bridge} is the instrumented bridge length.

2) Energy Decay Method: The energy decay approach uses the physical relationship between signal energy and distance from the impact source. Signal energy was calculated for each sensor over a 1-second window centered on the impact time:

$$E_i = \sum_t s_{corrected,i}^2(t) \quad (14)$$

The energy decay model assumes geometric spreading losses following:

$$E_i = E_0 \frac{1}{\left(\left|x_{impact} - x_{sensor,i}\right| + \delta\right)^n} \quad (15)$$

where E_0 is the source energy, δ is a small offset to avoid singularities, and n is the decay exponent (set to 2.0 for geometric spreading).

Impact position and source energy were estimated through nonlinear least-squares optimization in logarithmic space for numerical stability:

$$\left(\hat{x}_{impact}, \hat{E}_0\right) = \arg\min_{x,E_0} \sum_{i=1}^{N} \left[\ln(E_i) - \ln\left(E_{\text{model},i}(x, E_0)\right)\right]^2 \quad (16)$$

3) Hybrid Method: The hybrid algorithm combines TDOA and energy decay estimates through weighted averaging:

$$\hat{x}_{hybrid} = w_{TDOA}.\hat{x}_{TDOA} + w_{energy}.\hat{x}_{energy} \quad (17)$$

Weights were set to $w_{TDOA} = 0.7$ and $w_{energy} = 0.3$ based on preliminary analysis indicating higher reliability of timing-based methods for the experimental configuration.

E. Performance Evaluation

Algorithm performance was assessed using leave-one-test-out cross-validation to ensure unbiased evaluation. For each test configuration, localization algorithms were applied using wave speed estimated from the remaining three test cases.

Performance metrics included:

- Mean Absolute Error (MAE): $MAE = \frac{1}{N}\sum_{i=1}^{N}\left|x_{\text{actual},i} - x_{\text{predicted},i}\right|$
- Relative Error: Relative error = $\frac{MAE}{L_{bridge}} \times 100\%$
- Success Rate: Percentage of tests yielding valid localization estimates.
- Standard Deviation: Variability of localization errors across test cases

Parameter sensitivity analysis was conducted by varying key algorithm parameters (wave speed for TDOA, decay exponent for energy method) to assess robustness under different conditions.

IV. RESULTS

A. Impact Localization

The 4σ threshold-based detection algorithm achieved very successful impact identification across all test configurations. Detection performance showed clear proximity-dependent amplitude variations, with sensors closest to impact locations consistently showing maximum responses.

The 80kg (Figure 4) loading produced maximum response amplitudes of ±0.025 nm, while 40kg (Figure 5) loading generated reduced but clearly detectable responses up to ±0.015 nm. For Test_1 (2m impact), Ch01Gr02 (2m sensor) dominated with peak responses of -0.009 nm (80kg) and -0.007 nm (40kg). Similarly, Test_4 (8m impact) showed Ch01Gr04 (8m sensor) maximum responses of +0.024 nm (80kg) and +0.014 nm (40kg).

Background noise remained consistent at ±0.005 nm across all channels, ensuring reliable threshold establishment. Detection thresholds varied from 0.006 nm to 0.019 nm based on individual sensor baseline characteristics, with all channels successfully detecting impacts within their effective sensing range.

Impact timing estimation proved consistency across test configurations, with estimated times ranging from 5.244s to 6.638s. Multiple sensor detections within individual tests showed temporal agreement, validating synchronization accuracy and supporting subsequent time-difference-of-arrival calculations.

The detection algorithm showed robust performance across varying signal conditions with no false positive detections during baseline periods. All genuine impact events produced clear threshold exceedances, establishing reliable foundation for physics-based localization algorithms. The consistent spatial detection patterns across both loading conditions validate the sensor network design for structural monitoring applications.

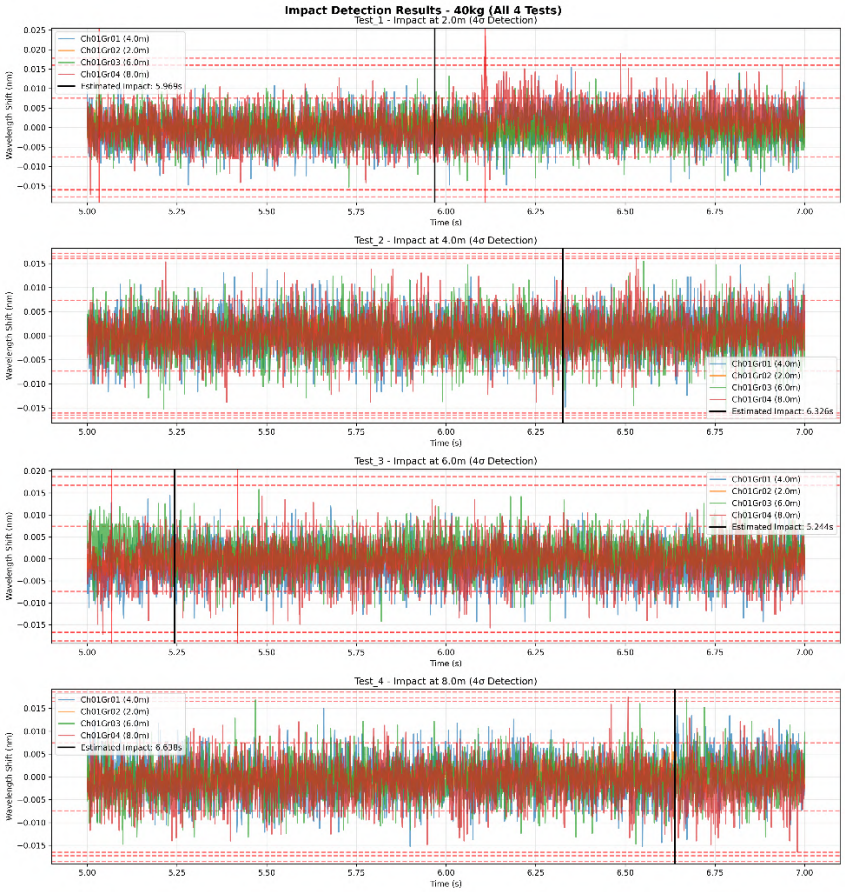


Figure 3: Impact detection results for 40kg loading showing baseline-corrected FBG sensor responses with detection thresholds (dashed red lines) and estimated impact times (vertical black lines)

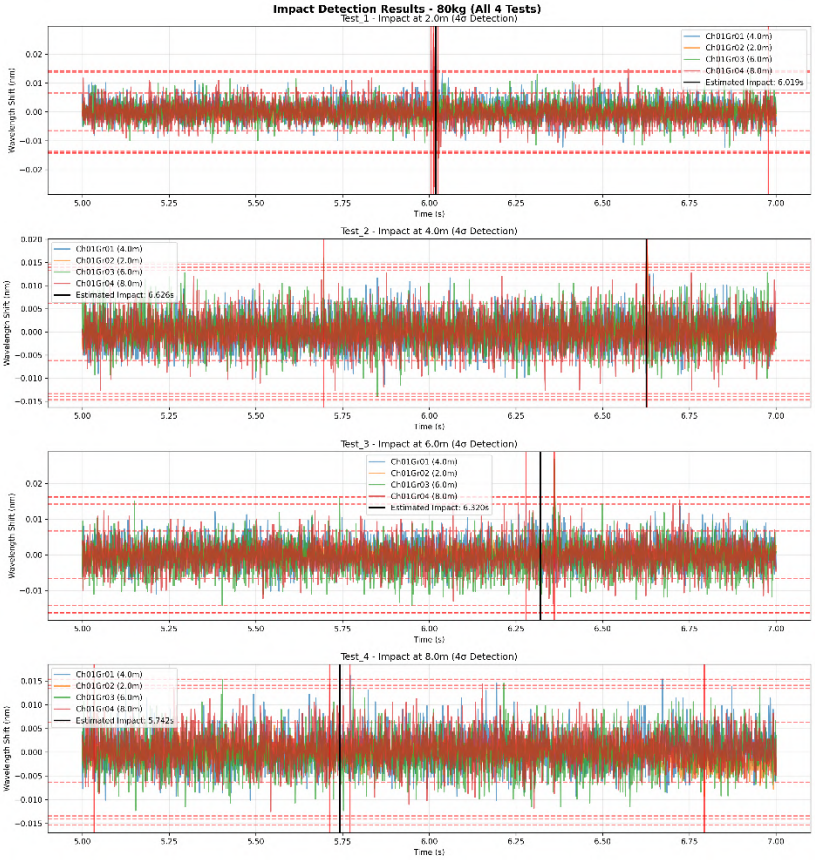


Figure 4: Impact detection results for 80kg loading showing baseline-corrected FBG sensor responses with detection thresholds (dashed red lines) and estimated impact times (vertical black lines)

B. Wave Speed Estimation Results

The analysis yielded a wave speed estimate of 508 m/s with a correlation coefficient of $R^2 = 0.611$, showing moderate but meaningful correlation between measured time differences and geometric distance relationships as presented on Figure 5.

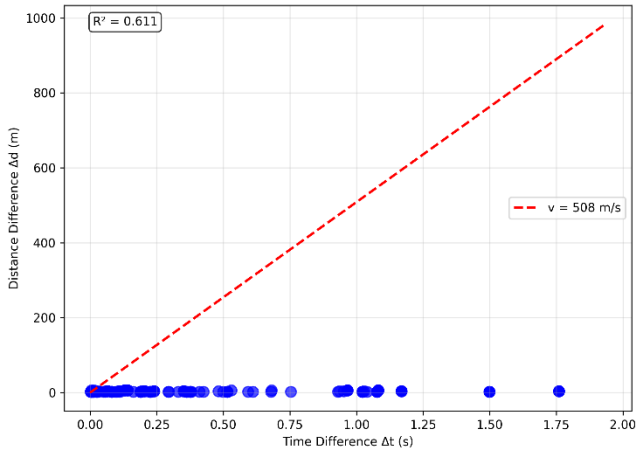


Figure 5 : Wave Speed Estimation Analysis

The wave speed of 508 m/s represents approximately 12.7% of the theoretical compression wave speed in concrete (≈4000 m/s), which is consistent with flexural wave propagation characteristics in beam-like structures [26]. This relationship aligns with established wave propagation theory, where flexural waves travel significantly slower than compression waves due to the coupling between bending and inertial effects in structural elements.

The correlation coefficient of $R^2 = 0.611$ demonstrates that the linear wave propagation model captures the primary physical relationship between time delays and sensor-impact distances, accounting for approximately 61% of the variance in the experimental data. This level of correlation is acceptable for practical structural monitoring applications, particularly considering the challenges associated with multi-mode wave propagation, material heterogeneity, and measurement uncertainties inherent in field conditions.

C. Algorithm Performance Comparsion

The 40kg loading condition reveals significant limitations for physics-based localization algorithms (Figure 6), particularly for TDOA methods which achieved 4.000m mean absolute error (40% relative error). This poor performance suggests insufficient signal amplitude for reliable arrival time determination across the sensor network. The energy decay method performed moderately better at 2.006m but exhibited systematic algorithmic bias, consistently predicting impact locations around 5.1m regardless of actual position.

The predictions versus actual position plot demonstrates this convergence behavior, with all energy decay predictions clustering tightly near the bridge center rather than following the diagonal trend line. This indicates the optimization algorithm struggles to differentiate spatial positions under reduced signal amplitudes. The error distribution shows significant scatter with frequent errors exceeding 3m, reflecting challenges in achieving precise localization under insufficient loading conditions. The hybrid method achieved marginal improvement at 2.149m, though overall performance remained suboptimal for practical monitoring applications.

The 80kg loading condition demonstrates optimal algorithm performance across all localization methods (Figure 7). The hybrid approach achieved excellent accuracy at 1.355m mean absolute error (13.5% relative error), representing suitable precision for structural health monitoring applications. Both TDOA (1.974m) and energy decay (1.971m) methods showed equivalent and improved performance compared to 40kg loading.

The predictions versus actual position plot reveals strong spatial discrimination capabilities, with localization estimates following the diagonal correlation line closely across all test positions. This improved correlation demonstrates that higher loading amplitudes provide optimal signal characteristics for physics-based algorithms. The error distribution concentrates below 2m, with most predictions achieving sub-2m accuracy. The bridge configuration clearly shows successful spatial resolution across all four impact positions, validating that adequate signal strength enables reliable localization performance suitable for real-world structural monitoring applications.

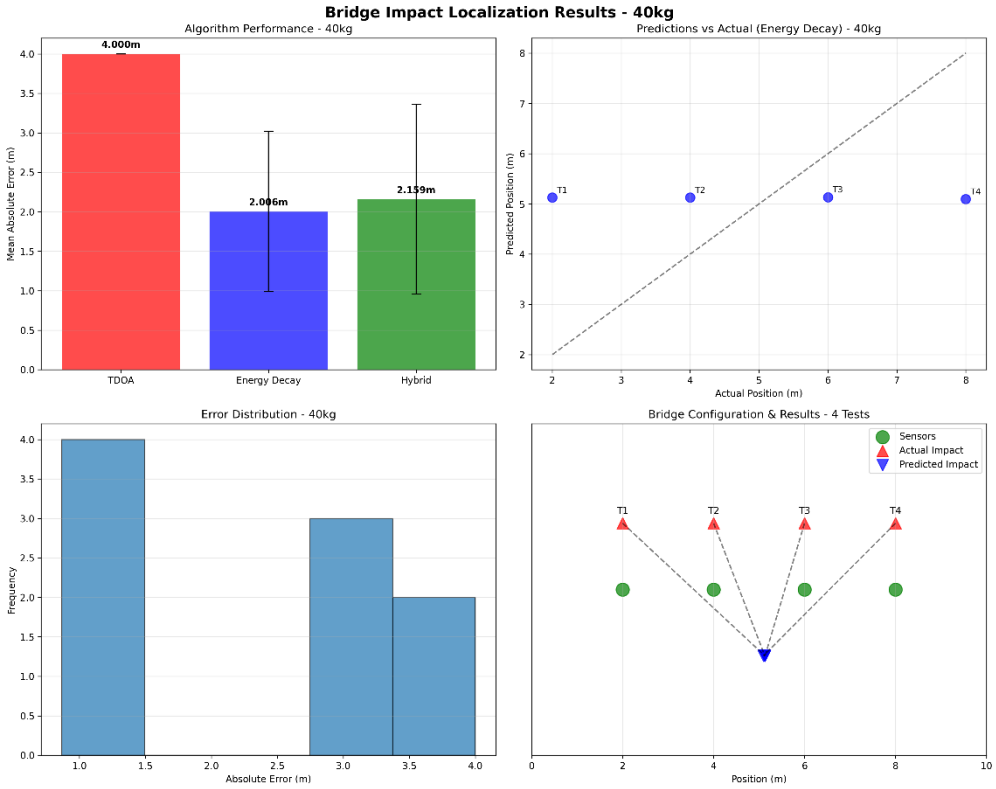


Figure 6: 40Kg Algorithm Performance

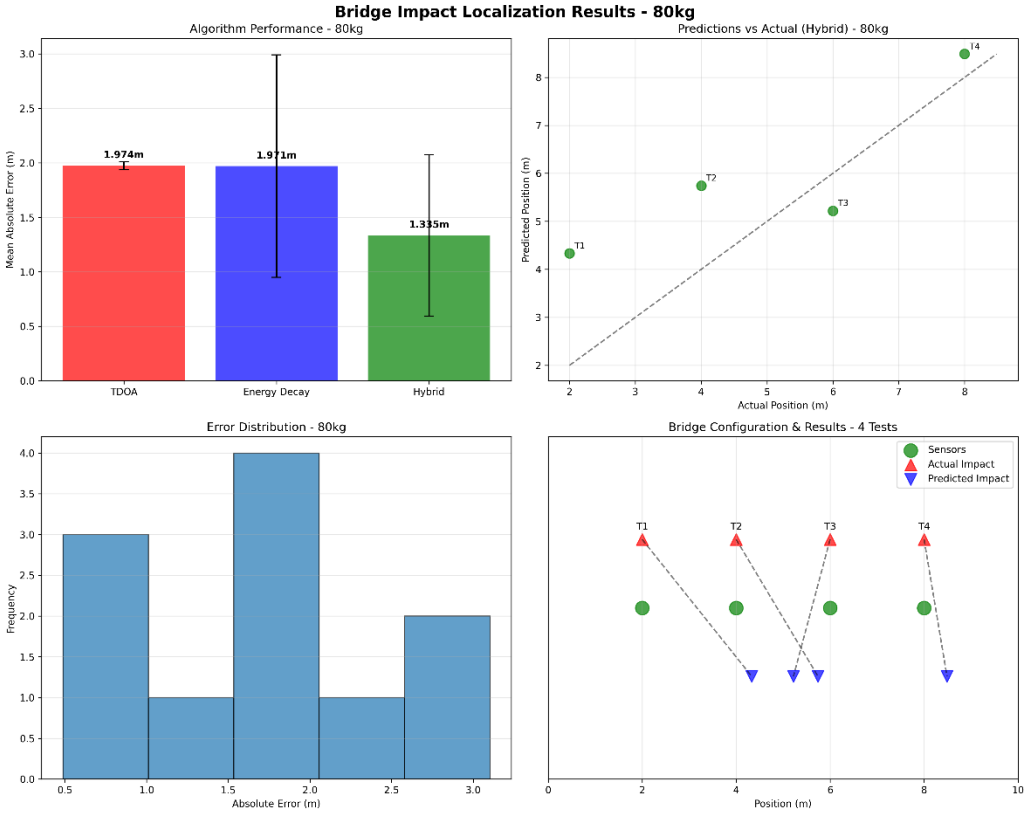


Figure 7: 80Kg Algorithm Performance

D. Algorithm Performance Analysis

Figure 8 demonstrates clear loading-dependent algorithm effectiveness, with 80kg conditions producing superior performance across all metrics compared to 40kg loading. The hybrid method achieved optimal results with 1.35m mean absolute error under 80kg loading versus 2.2m for 40kg conditions, while TDOA methods exhibited the greatest sensitivity with 1.97m (80kg) degrading to 4.0m (40kg) representing 100% performance loss under reduced loading. Success rate analysis reveals energy decay and hybrid algorithms maintaining 100% reliability across both conditions, while TDOA success dropped dramatically from 75% (80kg) to 25% (40kg), indicating fundamental challenges with arrival time determination at reduced signal amplitudes. Relative error performance shows the hybrid method achieving excellent 13.5% bridge-length accuracy under 80kg loading escalating to 22% under 40kg conditions, with energy decay methods maintaining consistent ~20% performance and TDOA varying from 20% to 40% across loading conditions. The complete 4/4 test coverage for both loading scenarios ensures statistical confidence in these performance comparisons, validating that adequate signal strength is critical for reliable physics-based localization in structural monitoring applications.

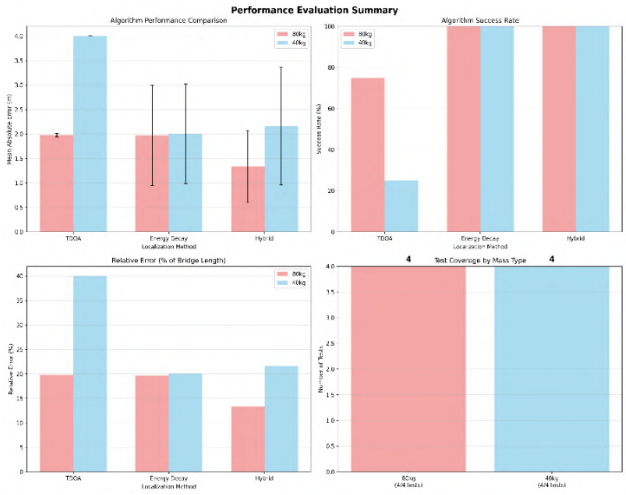


Figure 8: Summary of Algorithm Performance

V. Conclusion

This study successfully demonstrated the feasibility of physics-based impact localization using sparse FBG sensor networks for bridge monitoring applications. The experimental validation revealed significant loading dependent performance characteristics, with 80kg conditions enabling superior localization accuracy compared to 40kg loading across all algorithms. The hybrid method achieved optimal performance under adequate loading conditions with 1.35m mean absolute error (13.5% relative error), representing suitable precision for structural health monitoring applications. Wave speed estimation yielded 508 m/s with meaningful correlation (R^2 = 0.611), validating the fundamental assumptions underlying TDOA-based approaches. Energy decay methods demonstrated superior robustness with 100% success rates across loading conditions, while TDOA methods showed high sensitivity to signal amplitude variations. The results indicate that physics-based localization algorithms can achieve practical accuracy for real-world bridge monitoring when adequate impact energy ensures sufficient signal-to-noise ratios. Future work should investigate optimal loading thresholds, expanded sensor configurations, and algorithm adaptations for varying environmental conditions to enhance the reliability and applicability of FBG-based impact localization systems.

Acknowledgment

This work was supported by CSIR African Laser Centre, Telkom, and the National Laser Center (NLC).NK KAPINDU for his assistance in the compilation of this paper .

References

[1] C. L. Chen, C. C. W. Hung, and W. Y. Zhang, "Establishing severe deterioration and maintenance rules of bridge components: An empirical study in Taiwan," *Case Studies in Construction Materials*, vol. 22, p. e04907, Jul. 2025, doi: 10.1016/J.CSCM.2025.E04907.

[2] P. F. Van and D. Spuy, "The state of bridge loading in south africa," 2022, *Southern African Transport Conference*. Accessed: Sep. 21, 2025. [Online]. Available: https://repository.up.ac.za/handle/2263/87355

[3] K. Balasubramaniam, S. Sikdar, D. Ziaja, M. Jurek, R. Soman, and P. Malinowski, "A global-local damage localization and quantification approach in composite structures using ultrasonic guided waves and active infrared thermography," *Smart Mater Struct*, vol. 32, no. 3, p. 035016, Feb. 2023, doi: 10.1088/1361-665X/ACB578.

[4] Q. Chen, Z. Yang, X. Li, X. Sun, and Q. Kong, "Three-Dimensional Impact Localization on Concrete Structures Using Novel Enhanced Cross-Correlation Algorithm and Designed Concrete Implantable Module," *Front Mater*, vol. 9, p. 909006, Jun. 2022, doi: 10.3389/FMATS.2022.909006/BIBTEX.

[5] X. Huang, T. Peng, S. Wu, and X. Ming, "Efficient structural impact localization via signal curvature energy and probabilistic error function," *Measurement*, vol. 252, p. 117418, Aug. 2025, doi: 10.1016/J.MEASUREMENT.2025.117418.

[6] J. Braunfelds *et al.*, "FBG-Based Sensing for Structural Health Monitoring of Road Infrastructure," *J Sens*, vol. 2021, 2021, doi: 10.1155/2021/8850368.

[7] A. N. D. Alhussein, M. R. T. M. Qaid, T. Agliullin, B. Valeev, O. Morozov, and A. Sakhabutdinov, "Fiber Bragg Grating Sensors: Design, Applications, and Comparison with Other Sensing Technologies," *Sensors 2025, Vol. 25, Page 2289*, vol. 25, no. 7, p. 2289, Apr. 2025, doi: 10.3390/S25072289.

[8] S. Wang, Y. Xiang, Y. Gao, Y. Hua, C. Huang, and X. Zhao, "The Time Difference of Arrival Estimation Method Utilizing an Inexact Reconstruction Within the Framework of Compressed Sensing," *Remote Sensing 2024, Vol.*

16, Page 4039, vol. 16, no. 21, p. 4039, Oct. 2024, doi: 10.3390/RS16214039.

[9] M. Sun and K. C. Ho, "Energy-based source localization with non-ideal energy decay factor," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 2561–2564, 2008, doi: 10.1109/ICASSP.2008.4518171.

[10] N. V. Golovastikov, N. L. Kazanskiy, and S. N. Khonina, "Optical Fiber-Based Structural Health Monitoring: Advancements, Applications, and Integration with Artificial Intelligence for Civil and Urban Infrastructure," *Photonics 2025, Vol. 12, Page 615*, vol. 12, no. 6, p. 615, Jun. 2025, doi: 10.3390/PHOTONICS12060615.

[11] Z. Yang *et al.*, "A review in guided-ultrasonic-wave-based structural health monitoring: From fundamental theory to machine learning techniques," *Ultrasonics*, vol. 133, Aug. 2023, doi: 10.1016/j.ultras.2023.107014.

[12] I. Floris, J. M. Adam, P. A. Calderón, and S. Sales, "Fiber Optic Shape Sensors: A comprehensive review," *Opt Lasers Eng*, vol. 139, Apr. 2021, doi: 10.1016/j.optlaseng.2020.106508.

[13] C. Y. Hong, Y. F. Zhang, M. X. Zhang, L. M. G. Leung, and L. Q. Liu, "Application of FBG sensors for geotechnical health monitoring, a review of sensor design, implementation methods and packaging techniques," *Sens Actuators A Phys*, vol. 244, pp. 184–197, Jun. 2016, doi: 10.1016/j.sna.2016.04.033.

[14] S. Deepa and B. Das, "Interrogation techniques for π-phase-shifted fiber Bragg grating sensor: A review," *Sens Actuators A Phys*, vol. 315, Nov. 2020, doi: 10.1016/j.sna.2020.112215.

[15] A. Rostami *et al.*, "Advances in fibre Bragg grating technology for magnetic field sensing: A review," *Measurement (Lond)*, vol. 221, Nov. 2023, doi: 10.1016/j.measurement.2023.113482.

[16] L. Zhao, C. Bi, H. Huang, Q. Liu, and Z. Tian, "A review of acoustic Luneburg lens: Physics and applications," *Mech Syst Signal Process*, vol. 199, Sep. 2023, doi: 10.1016/j.ymssp.2023.110468.

[17] X. Liu, F. Xiong, Q. Xie, X. Yang, D. Chen, and S. Wang, "Research on the Attenuation Characteristics of High-Frequency Elastic Waves in Rock-Like Material," *Materials 2022, Vol. 15, Page 6604*, vol. 15, no. 19, p. 6604, Sep. 2022, doi: 10.3390/MA15196604.

[18] B. Ma, C. Tong, L. Zou, and M. Tian, "A TDOA localization method for complex environment localization," *J Phys Conf Ser*, vol. 2004, no. 1, Sep. 2021, doi: 10.1088/1742-6596/2004/1/012003.

[19] N. Zhu and T. Reza, "A modified cross-correlation algorithm to achieve the time difference of arrival in sound source localization," *Measurement and Control (United Kingdom)*, vol. 52, no. 3–4, pp. 212–221, Mar. 2019, doi: 10.1177/0020294019827977.

[20] W. Meng, W. Xiao, B. Jin, F.-B. Yang, S. Sun, and H. Wei, "Energy-Based Acoustic Source Localization Methods: A Survey," *Sensors 2017, Vol. 17, Page 376*, vol. 17, no. 2, p. 376, Feb. 2017, doi: 10.3390/S17020376.

[21] M. E. De Simone, F. Ciampa, S. Boccardi, and M. Meo, "Impact source localisation in aerospace composite structures," *Smart Mater Struct*, vol. 26, no. 12, p. 125026, Nov. 2017, doi: 10.1088/1361-665X/AA973E.

[22] A. Coles, B. A. de Castro, C. Andreades, F. G. Baptista, M. Meo, and F. Ciampa, "Impact Localization in Composites Using Time Reversal, Embedded PZT Transducers, and Topological Algorithms," *Front Built Environ*, vol. 6, p. 519441, Mar. 2020, doi: 10.3389/FBUIL.2020.00027/BIBTEX.

[23] K. Kim, J. Choi, J. Chung, G. Koo, I. H. Bae, and H. Sohn, "Structural displacement estimation through multi-rate fusion of accelerometer and RTK-GPS displacement and velocity measurements," *Measurement*, vol. 130, pp. 223–235, Dec. 2018, doi: 10.1016/J.MEASUREMENT.2018.07.090.

[24] J. Yu and D. Liang, "Impact localization system of composite structure based on recurrence quantification analysis by using FBG sensors," *Optical Fiber Technology*, vol. 49, pp. 7–15, May 2019, doi: 10.1016/J.YOFTE.2019.01.024.

[25] C. Duan *et al.*, "FBG array arrangement optimization for impact localization," *Meas Sci Technol*, vol. 35, no. 2, p. 025206, Nov. 2023, doi: 10.1088/1361-6501/AD0B69.

[26] Y. H. Lee and T. Oh, "The Measurement of P-, S-, and R-Wave Velocities to Evaluate the Condition of Reinforced and Prestressed Concrete Slabs," *Advances in Materials Science and Engineering*, vol. 2016, 2016, doi: 10.1155/2016/1548215.

IsiNdebele Question-Answering Model using Minimal Dataset

1st Promise Tshepiso Malatji
Department of Computer Science
University of Limpopo
Polokwane, South Africa
promise.malatji@ul.ac.za

2nd Thipe Isaiah Modipa
Department of Computer Science
University of Limpopo
Polokwane, South Africa
thipe.modipa@ul.ac.za

***Abstract*— The adoption of pre-trained Large Language Models (LLMs) using low resourced data has gain surge recently. With Question Answering (QA) being one of the specific tasks under Natural Language Processing (NLP), such a task is important in teaching and learning process. For resourced languages like English, QA datasets are readily available however there are no advances for low-resource languages like isiNdebele language. Amongst other challenges of fine-tuning a model with low-resource language includes overfitting and unstable optimization. A custom Nguni tokenizer is developed using SentencePiece model and used to train the multilingual text-to-text transfer transformer (mT5)-small architecture to developing an isiNdebele language model. The developed isiNdebele language model was fine-tuned for explicit and implicit QA. The implicit_QA dataset was augmented to enhance the model performance. Two zero-shot mT5 QA baseline models were developed using the same explicit_QA and augmented implicit_QA datasets. The implicit isiNdebele QA model trained on the augmented data scored ROUGE-L and BERTScore F1 of 0,9911 and 0,9969 respectively.**

Keywords—*question answering, Natural language processing, custom tokenizer, isiNdebele dataset, mT5.*

I. INTRODUCTION

Question Answering (QA) is one of the specific task under Natural Language Processing (NLP). Question answering is important in teaching and learning process. There are advances made in this regard for well-resourced languages like English. Multiple Choice question is an example of close-ended questions that has recently gained attention [1]. Data scarcity is the main challenge for model development and evaluation in many languages [2]. The notable QA model performance gap between high resourced and low resourced languages is still evident. Method such as few-shots learning are used to improve model performance. Just like Thai, there are many advances made in English QA however the opposite can be said about South African low resource languages [3]. The English QA datasets are readily available for model development but there is no advance for South African isiNdebele language. Large Language Models (LLMs) require enormous amount of data to effectively train and fine tune for task specific [4]. The use of pre-trained LLMs to train low resourced data has gain surge recently. Fine-tuning in low resource languages is data inefficient, resulting in low accuracy [5]. Some of the challenges of fine-tuning a model with low-resource language includes overfitting and unstable optimization. The multilingual text-to-text transfer transformer (mT5) model is fine-tuned for several task specific such as text summarization, question-answering, classification tasks and sentiment analysis [6], [7], [8]. The purpose of this study is to develop a QA model using the available minimal dataset augmented through exhaustive word-level masking. IsiNdebele language is considered a low-resourced language with minimal online presence [9].

This paper is arranged as follows, Section II discusses the related work on QA models for low resourced languages on small datasets. A detailed methodology that covers datasets used, model development and fine-tuning is covered in section III. Section IV presents the experimental results obtained. The results discussion is outlined in section V. Lastly, section VI gives a concise conclusion to the study.

II. RELATED WORK

Fine-tuning a pretrained model is retraining the model for a specific task such as Question-answering, summarization and translation [10]. Pretrained models like mT5, mBERT, GPT and XLM-R were fine-tuned and evaluated using a low resourced language Kazakh dataset [11]. The strategies used to improve the performance are hyperparameter optimization and data augmentation techniques like back-translation and paraphrasing [11]. The mT5 outperformed all the other models with an F1 score of 75.72%. The monolingual models like Kaz-RoBERTa underperformed compared to larger multilingual models. Amongst other challenges, the Kazakh's agglutinative morphology structure and data scarcity contributed to the performance [11]. Therefore, a conclusion that data augmentation and advanced fine-tuning are key in low resourced language QA model development was reached. Several LLMs were fine-tuned using six Kazakh datasets for QA, casual reasoning math problems, machine translation and spelling correction [12]. For classification tasks the LLMs performed better compared to generative tasks. The two models, XLM-RoBERTa and mBERT were fine-tuned using their custom KazQA dataset for extractive QA [13]. The KazQA dataset consists of 2000 question-answer pairs crawled from Wikipedia articles. The results obtained indicate the importance of using multilingual models for agglutinative languages and the need for domain-specific datasets. An Arabic extractive QA was developed using 43842 question-answer pairs that aimed at benchmarking for the Arabic language [14]. The Holy Qur'an was used in the development of a QA model to address religious understanding and queries [15]. An approach to develop multilingual QA using five examples per language to address data scarcity in low resource languages [16]. A few-shot learning setting is used to address the data scarcity by generating multilingual QA data that is used to fine-tune models [16].

A Thai QA model is developed using three sets of data, the Thai Wiki, iApp training datasets and generated data [17]. Fig. 1 illustrates the QA framework for developing the Thai QA models using the WangchamBERTa and mT5- Base architectures. The developed models were then evaluated

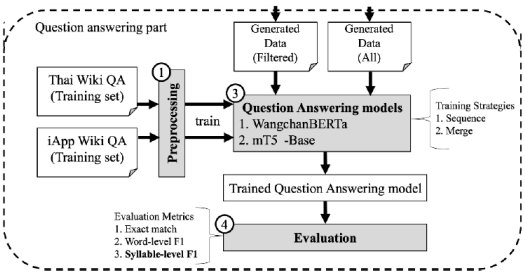


Fig. 1. The Thai QA framework depicting the flow of data in the development of QA models [17].

using the exact match, word-level F1 and syllable-level F1 metrics. There are two strategies that can be used to train the QA model with low data, the sequence and merge strategy [17]. With sequence strategy the datasets are used separately and one after the other while the merge strategy involves combining the different datasets before training the model.

III. Methodology

In this section we detail the steps followed in developing the two isiNdebele QA models, the isiNdebele explicit (Nbl-Exp) and isiNdebele implicit (Nbl-Imp) QA models and their respective baseline QA models.

A. Datasets

The datasets used in this study was obtained from the National Centre for Human Language Technology (NCHLT)[1] and the South African Department of Basic Education (DBE) [2] websites. For tokenizer and model development the Nguni family languages (isiNdebele, isiZulu, isiXhosa, and siSwati) and the isiNdebele datasets were used respectively. The Nguni dataset is a combination of two isiNdebele, two isiZulu, and two isiXhosa datasets and one siSwati dataset with 285815 sentences. The isiNdebele dataset has 151422 sentences from the two datasets used. Autoencoding technique was used where each sentence serves as the input and its respective output to form the pair for model training format. The isiNdebele dataset was divided into 70:10:20, for training, validation and testing.

For fine-tuning the model, the DBE grade 12 papers from the National Senior Certificate (NSC) examinations were used in the creation of the datasets. A total of 45 isiNdebele question papers and their respective memorandums were sourced from the DBE. The focus was only on paper 1 comprehension questions that contain a text and set of questions based on the given context. Some questions and answers share the same context as they are derived from the same question paper. The dataset is named explicit_QA and contains three columns: context, question, and answer. The resulting dataset has 1276 triplets to be used for the explicit QA model training and testing. To create the implicit dataset named implicit_QA, all the 45 contexts were combined to form the new dataset. The dataset was then split into sentences. The resulting dataset with 3032 sentences was divided into training, validation and testing using the 70:10:20 ratio, 2121 ,304 and 607 respectively. A random word was then masked from each sentence. The sentence with a masked word is the question, and the missing (masked) word is the answer. The implicit_QA dataset consists of sets of triplets, the context which is the complete sentence, the question and answer. To create an augmented implicit_QA

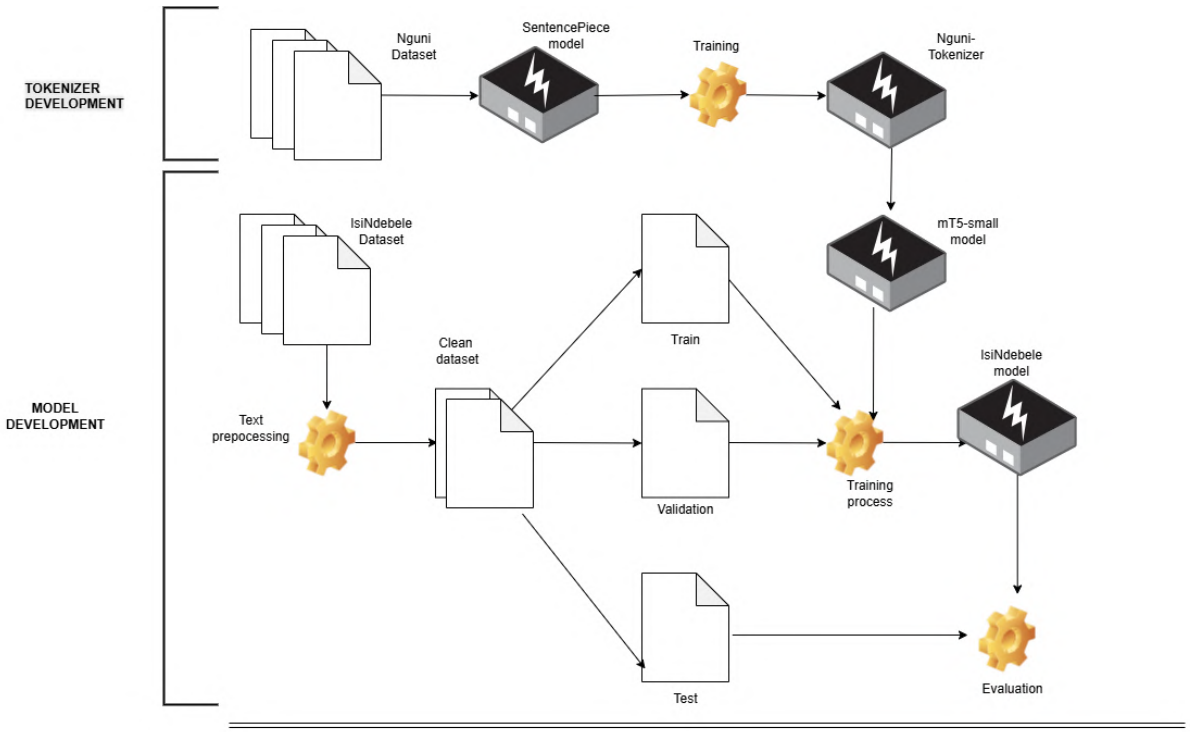


Fig. 2. The tokenizer and model development stages depicting the NguniTokenizer development and usage in the model training process.

[1]https://repo.sadilar.org/collections/dca359ff-fd5d-4d79-a957-b7bc3cb75bc9

[2]https://www.education.gov.za/Curriculum/NationalSenioCertificate(NSC)Examinations/NSCPastExaminationpapers.aspx

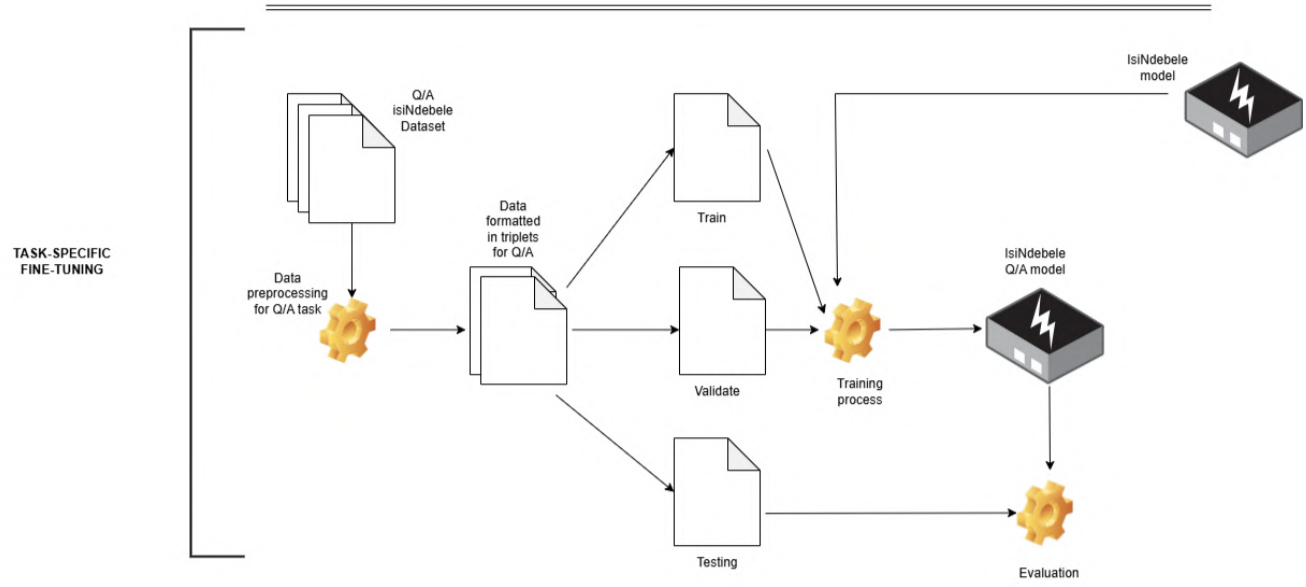


Fig. 3. The fine-tuning process for developing the explicit and implicit QA models using the isiNdebele model.

(augImpicit_QA) dataset, each word in every sentence was masked. The developed augImpicit_QA dataset consist of 29331 training and validation triplets from the 2425 original training and validation sentences. From 607 testing sentences the augmentation process yielded 7229 testing triplets. The augmentation significantly increased the implicit_QA dataset from 3032 triplets to 36560. This also improves the word coverage since each word in the dataset served as the target word.

B. Model Development

The initial task was to develop a customized Nguni tokenizer (NguniTokenizer) using the Nguni dataset. The SentencePiece model architecture was used with the unigram model type. The SentencePiece model receives raw Nguni dataset as its input and develop a NguniTokenizer by subword-level tokenizing the dataset to produce list of subword tokens as output. The vocabulary size used is 120000 and the output is the NguniTokenizer as illustrated in Fig. 2, the tokenizer development stage. The default MT5Tokenizer was replaced with the custom NguniTokenizer in the mT5-small to develop the isiNdebele language model named NguniTok-nbl model. The learning rate of 3e-5, and batch size x gradient accumulation of 4 x 2 (total of 8) were used. The number of encoder and decoder layers used are eight and the attention layers used are six. Fig. 2, the model development stage demonstrates the graphical flow of the model development from the dataset, text preprocessing and split to model training and evaluation.

C. QA Models

The developed isiNdebele model was fine-tuned for task specific, Question-Answering (QA). Fig. 3 depicts the fine-tuning process, from the QA dataset to the final QA model. Two QA models were developed using the two QA datasets, explicit_QA and implicit_QA. After preparing the datasets, we then used the isiNdebele model as illustrated in Fig. 3 and the QA model is evaluated using the testing dataset.

1) Explicit QA

The developed NguniTok-nbl model was then retrained for QA using the explicit_QA dataset. Several experiments were conducted, the first experiment was the batch size where 8 experiments were conducted for different batch x gradient accumulation combinations. The results were then used to select the best performing model to be further fine-tune. The effect of the learning rate was also investigated through slowly incrementing the learning rate from 1e-5 to 8e-5. The final experiment was on the learning rate scheduler, with three schedulers investigated, the linear, cosine and polynomial. The optimal model is named Nbl-Exp. Table I illustrates the different parameters of the Nbl-Exp and Nbl-Imp QA models and the baseline mT5 QA models. The model, tokenizer and language covered by the model tokenizer.

2) Implicit QA

Using the Implicit_QA dataset three experiments were conducted to develop and optimize the implicit QA model, named Nbl-Imp. The first experiment is based on the different batch sizes, 2x2, 4x2 and 8x2. The second one is monitoring the increase in learning rate from 1e-5, 3e-5, 5e-5 to 8e-5.

TABLE I. THE PARAMETERS USED IN THE DEVELOPMENT OF THE QA MODELS.

QA Model	Parameters			
	Model	Tokenizer	Lang Covered	Dataset
Nbl-Exp	NguniTok-nbl	Nguni	4	Explicit_QA
mT5-Exp	mT5-small	MT5	101	Explicit_QA
Nbl-Imp	NguniTok-nbl	Nguni	4	Implicit_QA
Nbl-AugImp	NguniTok-nbl	Nguni	4	Augmented Implicit_QA
mT5-Imp	mT5-small	MT5	101	Augmented Implicit_QA

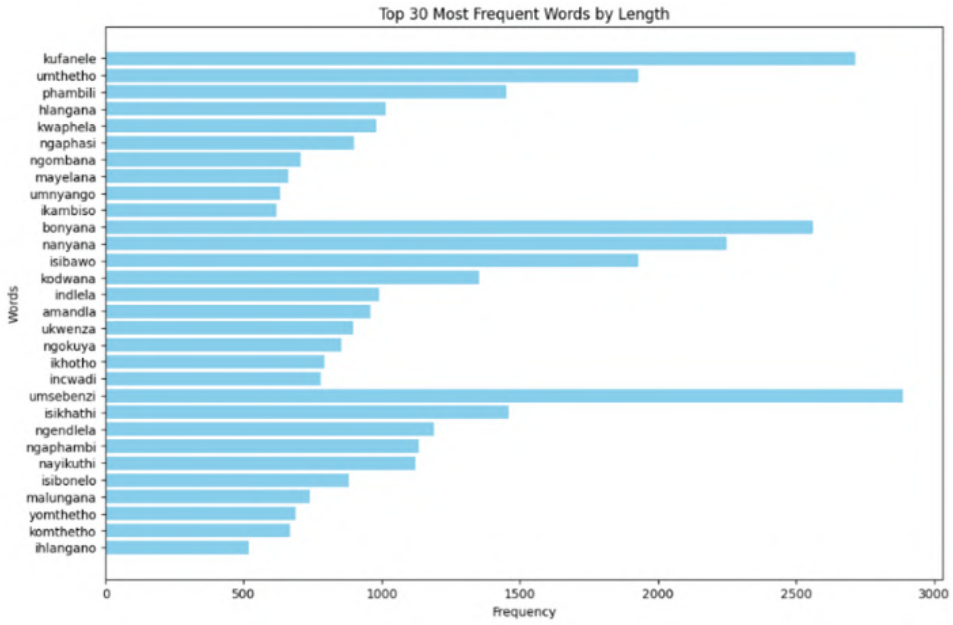


Fig. 4. The most frequent words in the isiNdebele dataset for the three highest word lengths 8, 7, and 9 respectively.

3) Augmented Implicit QA

Some experiments were performed using the AugImpicit_QA dataset to develop the isiNdebele augmented implicit (Nbl-AugImp) QA model. Since the batch size proved to be effective in small datasets. Nine batch size x gradient accumulation experiments (2x2, 2x4, 2x8, 4x2, 4x4, 4x8, 8x2, 8x4 and 8x8) were performed. We further trained the model with four different learning rates, 1e-5, 3e-5, 5e-5 and 8e-5.

4) Baseline QA models

Two zero-shot QA models were developed using the mT5-small model. The first baseline QA model is the mT5-Exp QA model which is developed based on the Nbl-Exp QA model optimal parameters. The second baseline QA model is the mT5-Imp QA model developed using the AugImplicit_QA dataset. The implicit zero-shot model was developed using the optimal parameters used to obtain the Nbl-AugImp model.

5) Evaluation

	A	B	C
1	question	context	answers
2	ngisiphi isifundo ekumele sifundiswe eenkolweni esizakwenza bonyana abafundi bakghone ukukhetha ibizelo elihle lomsebenzi?	abantu abanengi esewula afrika,	isifundo se-career guidance.
3	abantu abatjha kufuze bakhethe nini ibizelo labo lomsebenzi?	abantu abanengi esewula afrika,	kufuze bakhethe nabaqeda noma nabenza ibanga letjhumi noma igreyidi le-12.
4	kubayini umuntu afanele akghodlhelele ebizelweni lakhe ngemva kokulikhetha?	abantu abanengi esewula afrika,	kumele akghodlhelele ngombana woke amabizelo anobudisi nobumnandi bakhona. ayikho into etholakala lula ephasinapha.

Fig. 5. The explicit QA data sample of question and answers based on the same context.

The developed isiNdebele model and the QA models were evaluated using several matrices. The Recall-Oriented Understudy for Gisting Evaluation – Long sequence of words (ROUGE-L) focuses on the lexical similarity of the generated text to the reference text [10]. The transformer based metric, Bidirectional Encoder Representations from Transformers score (BERTScore) F1 is used to measure the semantic similarity [18]. The Bilingual Evaluation Understudy (BLEU) measures the text quality. Perplexity measure the language model performance [19].

IV. Results

A. Data Analysis

Common words exist between the four Nguni languages, symbolising the morphological similarity that exists between them. Words such as *kanye* (once), *umuntu* (person), *abantu* (people), *khona* (there), *kufanele* (must) are similar in the datasets. The word length distribution is also similar with the mode of eight. Fig. 4 shows the most used words of length 8, 7 and 9 in the isiNdebele dataset. Common words include *kwenza*, *ikhoto* and *ihlangano* which are to-do, court, and part respectively, which are in line with the government domain where it was obtained. A sample of the explicit_QA dataset is illustrated in Fig. 5, where the answer is derived from the context. The context will be the same for all the questions and answers derived from it.

The AugImplicit_QA dataset sample as depicted in Fig. 6 has five generated fill-in questions from a single sentence. The training dataset had 29331 triplets (complete sentence, sentence with masked word and the masked word). The testing

1	context	question	answer
26	ukungabekezeli bese usebenzisa indlela erathako.	____ bese usebenzisa indlela erathako.	ukungabekezeli
27	ukungabekezeli bese usebenzisa indlela erathako.	ukungabekezeli ____ usebenzisa indlela erathako.	bese
28	ukungabekezeli bese usebenzisa indlela erathako.	ukungabekezeli bese ____ indlela erathako.	usebenzisa
29	ukungabekezeli bese usebenzisa indlela erathako.	ukungabekezeli bese usebenzisa ____ erathako.	indlela
30	ukungabekezeli bese usebenzisa indlela erathako.	ukungabekezeli bese usebenzisa indlela ____.	erathako

Fig. 6. The augmented implicit QA data samples of five fill-in sentences generated from one sentence.

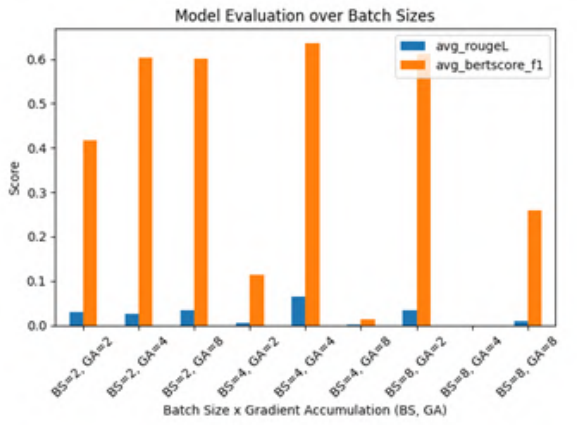


Fig. 7. The batch sizes x gradient accumulation results.

dataset had 7229 triplets. The augmentation significantly increased the dataset from 3032 to 36560 linguistically rich triplets with each word appearing as the target word.

B. *IsiNdebele Model*

The developed isiNdebele model performed well when measure using the four metrices, ROUGE-L, BERTScore, BLEU and perplexity. For lexical and semantical similarities, the model scored a ROUGE-L of 0.9629 and BERTScore F1 of 0.9904. The model performs well with a BLEU score of 0.8869, meaning it generate highly similar texts. An average perplexity of 1.0013 was achieved by the model, simplifying the ability of the model to make good predictions.

C. *Explicit QA*

The batch size experiment using the explicit_QA dataset yielded the results presented in Fig. 7. The batch sizes 4 x 2, 4 x 8 and 8 x 4 yielded the lowest BERTScore F1 and scores of 0.113156, 0.012518 and 0, respectively. The respective ROUGE-L were also lower than all other experiments. From the experiment, the model of batch size of 4 x 4 has a ROUGE-L and BERTScore F1 of 0.063947 and 0.636139, respectively. And the BERTScore F1 is the highest score in all the different batch size experiments. The model that outperformed the others in the batch size experiments was retrained on eight different learning rates with all other hyperparameters kept constant. The learning rate's impact was monitored and recorded based on the model performance. The model performance was lower at the learning rates 4e-5 and 8e-5. Learning rate 3e-5 proved to provide a slight improvement in the BERTScore F1 from 0.636139 to 0.636629, however there was a small decrease in the ROUGE-L score from 0.063947 to 0.059253. Given the performances of the other models, the

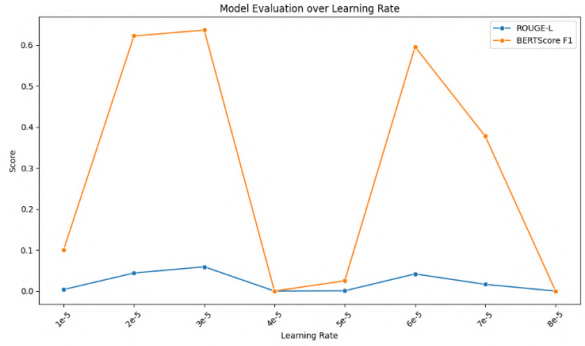


Fig. 8. The results for learning rate experiments.

TABLE II. THE BATCH SIZE RESULTS OF THE IMPLICIT QA MODEL.

Batch size	ROUGE-L	BERTScore F1
2x2	**0,991112187**	**0,996902053**
2X4	0,990032893	0,996842681
4X4	0,978695366	0,994070865
8X4	0,963688072	0,990515134

optimal model is the model trained with the learning rate of 3e-5 as illustrated in Fig. 8.

With the slight improved performance achieved with a learning rate of 3e-5, the next experiment focused on the effect of the learning rate scheduler. The default cosine was compared with the linear and polynomial models. The results presented in Fig. 9 demonstrates that the polynomial scheduler outperformed the other two models with a ROUGE-L score of 0.060378 and the BERTScore F1 of 0.658348. The linear learning rate scheduler performed the least with ROUGE-L and BERTScore F1 of 0.016784 and 0.489581.

D. *ImplicitQA*

The implicit QA model achieved performance great ROUGE-L and BERTScore F1 scores. When using the smallest batch size of 2x2, a ROUGE-L and BERTScore F1 of 0.596603 and 0.856168. Increasing the batch size to 4x2 decreased the score to 0.094547304 for ROUGE-L and 0.656108955 for BERTScore F1. The model performance increase as the learning rate increases, with learning rate of 8e-5 the model performance was 0.905196241 and 0.975967097 for ROUGE-L and BERTScore F1 respectively.

E. *Augmented Implicit QA*

In the second implicit experiment, we fixed all the parameters and used 5e-5 as our learning rate. Four batch size experiments were conducted as shown in Table II. The largest batch size 8x4 achieved good results of 0.963688072 and

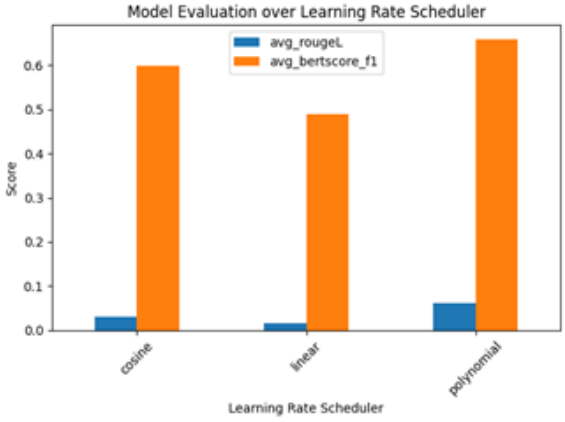


Fig. 9. The results for the three different learning rate schedulers.

TABLE III. THE COMPARISON EVALUATION RESULTS OF NBL-EXP QA MODEL AND THE BASELINE MT5-EXP QA MODEL.

	Explicit QA Models	
	Nbl-Exp	mT5-Exp
ROUGE-L	**0.060378**	0,019348536
BERTScore F1	**0.658348**	0,327261541
Perplexity	**8918.8682**	2438435.5000
BLEU Score	**0.0312**	0.0000

0.990515134 for ROUGE-L and BERTScore respectively. Reducing the batch size by half from 32 (8x4) to 16 (4x4) further improved the performance slightly. The batch size 8 (2x4) model recorded a ROUGE-L and BERTScore F1 of 0.990032893 and 0.996842681 respectively. The batch size 2x2 outperformed the other three models. The highest scores obtained in all the fill-in QA models are 0.991112187 and 0.996902053 for ROUGE-L and BERTScore F1 respectively. The results indicate that for small datasets better results can achieve by using small batch sizes.

F. Baseline QA models

The Nbl-Exp QA model was compared with the mT5-Exp QA model, and the results are presented in Table III. The Nbl-Exp QA model outperformed the zero-shot mT5 model with the same explicit dataset [20]. Both models performed poorly on ROUGE-L evaluation with 6.04% and 1.93% for Nbl-Exp and mT5-Exp QA model respectively. The BERTScore F1 for Nbl-Exp QA model is above 60%, indicating an average performance in terms of semantic similarity. A very high perplexity was recorded for both models, with the mT5-Exp QA scoring 2438435.5. Generally, the two models did not perform well on explicit QA.

TABLE IV. THE COMPARATIVE RESULTS OF NGUNITOK-IMP QA MODEL AND THE BASELINE MT5-IMP QA MODEL.

	Implicit QA Models	
	Nbl-AugImp	MT5-Imp
ROUGE-L	0,991112187	**0,992253424**
BERTScore F1	0,996902053	**0,998969181**
Perplexity	**1.145143**	1,156455083
Accuracy	**0.994467**	0,947441217
Precision	**1.0**	**1.0**
Recall	**0.994467**	0,947441217
F1 Score	**0.997226**	0,973011364

On implicit QA models, the performance significantly increased for both the Nbl-AugImp and mT5-Imp QA models. Since the answers to this type of questions are exact, we used metrics accuracy, precision, recall and F1 score to evaluate the performance of the models. The Nbl-AugImp QA model performed exceptionally well syntactically and semantically on the testing dataset. The mT5-Imp QA model outperformed the Nbl-AugImp QA model on the ROUGE-L and BERTScore F1 as depicted in Table IV. The Nbl-AugImp QA model generated texts which had the lowest average perplexity of 1.145143. Both models scored a precision level of 1.0 indicating that the models predict positive quality texts.

V. DISCUSSION

The use of the custom Nguni tokenizer in developing an isiNdebele language model proved to be effective and produced good results. The model had a ROUGE-L and BERTScore F1 of 96.29% and 99.04% respectively, meaning it generates syntactically and semantically text that is related to the target text. We then fine-tuned the model for QA, using two different question styles, the explicit and implicit. The explicit QA model and the mT5 baseline explicit model performed badly, with low ROUGE-L and BERTScore F1 scores. This poor performance is due to the small dataset used to fine-tune the models. For these explicit QA models, the total dataset had 1276 triplets, which is extremely small to effectively train and test a language model. Table V denotes samples of text predictions made by the Nbl-Exp and the mT5-Exp QA models. For the explicit QA, the sample question is "*sekela isitatimende esingenzasi ngephuzu elilodwa. kusisenzo esilungileko ukuphana ngezitho zomzimba."* translated to "support the statement below with one point. it is a righteous act to donate organs.". Both models performed badly as the Nbl-Exp QA model's response is part of the question while the mT5-Exp QA model's response is an extract from the context. These types of responses are observed throughout the evaluation process where both models return a part of the question or context as the answer because of lack of contextual coverage. And this justifies the poor lexical and syntactic score while there is an average performance in semantics. The predictions made are semantically related to the target text as they are drawn from the question or context.

While on the implicit models, the Nbl-Imp QA model achieved promising results after performing several experiments. With an implicit QA dataset, the Nbl-Imp QA model performed good, to optimize the results data augmentation was performed. Making each word in every sentence a target word significantly improved the model's performance. This broadens the word coverage and makes the model learn efficiently. The Nbl-AugImp QA model outperformed the mT5-Imp QA model on perplexity, accuracy, recall and F1 score. The performance difference of the two models is however less significant in terms of score values. For ROUGE-L and BERTScore F1, the Nbl-AugImp QA model scored 0.991112187 and 0.996902053 while the mT5-Imp QA model scored 0.992253424 and 0.998969181 respectively Table V consists of two samples of the fill-in question and their respective generated responses. From the two examples we note that the two models made correct predictions, this implies that the developed isiNdebele

TABLE V. THE GENERATED RESPONSES FOR ALL THE EXPLICIT AND IMPLICIT QA MODELS.

Models		Predictions	
Explicit QA	**Question**	*sekela isitatimende esingenzasi ngephuzu elilodwa. kusisenzo esilungileko ukuphana ngezitho zomzimba.* (support the statement below with one point. it is a righteous act to donate organs.)	
	Target Answer	**Nbl-Exp**	**mT5-Exp**
	Sisenzo esilungileko ngombana isithweso siyakwazi ukusindisa ipilo yomunye umuntu ukwenzela bona ipilwakhe iragele phambili. (It is the right thing to do because the gift can save someone's life so that his life can continue.)	*dwa. kusisenzo esilungileko ukuphana ngezitho zomzimba.* (month. it is a righteous act to donate one's organs.)	*matayere neensimbi eziphelileko ziyajanyiselelwa. Ikoloyi ithome iphile tswe! Ezinye zeensimbezi akusingezitja kodwana ngezakhutjhwa eenkoloyini ezalimalako nofana ezingasasebenzi* (worn-out tires and metals are replaced. The car starts alive! Some of the tears are not new but were removed from damaged or decommissioned vehicles)
Implicit QA			
	Question 1	*Alikho ithuba lokukhwabanisa amavowudu ngombana ____ iinhlangano zepolotiki nazo nanzi.* (There is no chance of vote rigging because ____ political parties are here too.)	
	Target Answer 1 *nakubalwako* (when counted)	**Nbl-Imp** *Nakubalwako* (when counted)	**mT5-Imp** *Nakubalwako* (when counted)
	Question 2	*Ngomnyaka we-1990 usosayensi wekhomphyutha uTim Berners-____ wasungula iWorld Wide Web.* (In 1990 computer scientist Tim Berners-____ invented the World Wide Web.)	
	Target Answer 2 *Lee* (Lee)	**Nbl-Imp** *Lee* (Lee)	**mT5-Imp** *Lee* (Lee)

model is performing well when fine-tuned for implicit QA than explicit QA.

VI. CONCLUSION

In this paper we presented the development of isiNdebele language model using a custom Nguni tokenizer. With the limited data in the Nguni languages, using the custom tokenizer proved to be beneficial for developing language models for low resourced languages. The developed model showed syntactical and semantical potential as it scored high in ROUGE-L and BERTScore F1. Though the model performed badly when fine-tuned for explicit QA, it did exceptionally well in implicit QA task. The Nbl-AugImp QA model achieved excellent results with augmented dataset to cover more target words. The results obtained were compared with a zero-shot mT5 QA models on the same training and testing dataset. The Nbl-AugImp QA model outperformed the mT5 QA model on the same dataset. The Nbl-ExpQA model did not generate syntactically correct predictions and that is due to the small dataset. Future work involves increasing the explicit QA dataset either by data collection or augmentation.

REFERENCES

[1] P. A. Atmajaya, A. Romadhony, and A. F. Ihsan, "Multiple-Choice Question Generation in Low Resource Indonesian Language with T5.," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 5, 2025.

[2] [A. Moreno-Cediel, J.-A. del-Hoyo-Gabaldon, E. Garcia-Lopez, A. Garcia-Cabot, and D. de-Fitero-Dominguez, "Evaluating the performance of multilingual models in answer extraction and question generation," *Sci. Rep.*, vol. 14, no. 1, p. 15477, 2024.

[3] N. Wiwatbutsiri, A. Suchato, P. Punyabukkana, and N. Tuaycharoen, "Question generation in the thai language using MT5," presented at the 2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE), IEEE, 2022, pp. 1–6.

[4] M. Szep, D. Rueckert, R. von Eisenhart-Rothe, and F. Hinterwimmer, "A Practical Guide to Fine-tuning Language Models with Limited Data," Nov. 14, 2024, *arXiv*: arXiv:2411.09539. doi: 10.48550/arXiv.2411.09539.

[5] T. Bansal, R. Jha, T. Munkhdalai, and A. McCallum, "Self-Supervised Meta-Learning for Few-Shot Natural Language Classification Tasks," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, 2020, pp. 522–534. doi: 10.18653/v1/2020.emnlp-main.38.

[6] F. Liu, T. Huang, S. Lyu, S. Shakeri, H. Yu, and J. Li, "EncT5: A Framework for Fine-tuning T5 as Non-autoregressive Models," Oct. 23, 2022, *arXiv*: arXiv:2110.08426. doi: 10.48550/arXiv.2110.08426.

[7] O. G. Toledano-López, J. Madera, H. González, A. Simón-Cuevas, T. Demeester, and E. Mannens, "Fine-tuning mT5-based Transformer via CMA-ES for Sentiment Analysis.," presented at the IberLEF@ SEPLN, 2022.

[8] V. N. Mahmoodabadi and F. Ghasemian, "Persian Text Summarization via Fine Tuning mT5 Transformer," *language*, 2023.

[9] J. O. Alabi, D. I. Adelani, M. Mosbach, and D. Klakow, "Adapting Pre-trained Language Models to African Languages via Multilingual Adaptive Fine-Tuning," Oct. 18, 2022, *arXiv*: arXiv:2204.06487. doi: 10.48550/arXiv.2204.06487.

[10] J. Becker, J. P. Wahle, B. Gipp, and T. Ruas, “Text Generation: A Systematic Literature Review of Tasks, Evaluation, and Challenges,” Aug. 29, 2024, *arXiv*: arXiv:2405.15604. doi: 10.48550/arXiv.2405.15604.

[11] A. Tleubayeva and A. Shomanov, “COMPARATIVE ANALYSIS OF MULTILINGUAL QA MODELS AND THEIR ADAPTATION TO THE KAZAKH LANGUAGE,” *Sci. J. Astana IT Univ.*, pp. 89–97, Sep. 2024, doi: 10.37943/19WHRK2878.

[12] A. Maxutov, A. Myrzakhmet, and P. Braslavski, “Do LLMs speak Kazakh? A pilot evaluation of seven models,” presented at the Proceedings of the First Workshop on Natural Language Processing for Turkic Languages (SIGTURK 2024), 2024, pp. 81–91.

[13] M. Shymbayev and Y. Alimzhanov, “Extractive Question Answering for Kazakh Language,” in *2023 IEEE International Conference on Smart Information Systems and Technologies (SIST)*, Astana, Kazakhstan: IEEE, May 2023, pp. 401–405. doi: 10.1109/SIST58284.2023.10223508.

[14] H. Al-Omari and R. Duwairi, “So2al-wa-Gwab: A New Arabic Question-Answering Dataset Trained on Answer Extraction Models,” *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 22, no. 8, pp. 1–21, Aug. 2023, doi: 10.1145/3605550.

[15] A. Sleem, E. M. lotfy Elrefai, M. M. Matar, and H. Nawaz, “Stars at qur’an qa 2022: Building automatic extractive question answering systems for the holy qur’an with transformer models and releasing a new dataset,” presented at the Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur’an QA and Fine-Grained Hate Speech Detection, 2022, pp. 146–153.

[16] P. Agrawal *et al.*, “QAMELEON : Multilingual QA with Only 5 Examples,” *Trans. Assoc. Comput. Linguist.*, vol. 11, pp. 1754–1771, Dec. 2023, doi: 10.1162/tacl_a_00625.

[17] P. Phakmongkol and P. Vateekul, “Enhance text-to-text transfer transformer with generated questions for Thai question answering,” *Appl. Sci.*, vol. 11, no. 21, p. 10267, 2021.

[18] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating Text Generation with BERT,” Feb. 24, 2020, *arXiv*: arXiv:1904.09675. doi: 10.48550/arXiv.1904.09675.

[19] A. Lavie and A. Agarwal, “Meteor: an automatic metric for MT evaluation with high levels of correlation with human judgments,” in *Proceedings of the Second Workshop on Statistical Machine Translation - StatMT '07*, Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 228–231. doi: 10.3115/1626355.1626389.

[20] W. Wongso, H. Lucky, and D. Suhartono, “Pre-trained transformer-based language models for Sundanese,” *J. Big Data*, vol. 9, no. 1, p. 39, Dec. 2022, doi: 10.1186/s40537-022-00590-7.

Comparing Video Classification Networks in Isolated Sign Language Recognition

1st Luke Goodall
Computer Science Department
Rhodes University
Makhanda, South Africa
0009-0001-7821-3343

2nd James Connan
Computer Science Department
Rhodes University
Makhanda, South Africa
0009-0005-9728-6192

3rd Dane Brown
Computer Science Department
Rhodes University
Makhanda, South Africa
0000-0001-7395-7370

4th Marc Marais
Computer Science Department
Rhodes University
Makhanda, South Africa
0000-0002-6952-4406

***Abstract*—Sign languages serve as the primary communication medium for millions worldwide, yet automatic translation systems remain limited compared to spoken language alternatives. This creates barriers to digital inclusion and accessibility. The use of 3D convolutional neural networks (CNNs) as visual backbones is well established in sign language recognition (SLR). However, vision transformers, which have achieved excellent results in broader computer vision tasks, have been less extensively explored for SLR-specific challenges. This gap can be partially attributed to the historical computational advantage of CNNs over transformer architectures, though recent vision transformer variants have achieved comparable efficiency. To address this gap, this study presents a systematic comparison of video classification architectures for isolated SLR, with particular attention to computational efficiency for resource-constrained scenarios. We trained and evaluated multiple architectures based on 3D convolution and vision transformers—including MViT, Swin3D, ResNet variants, and S3D—using the WLASL100 dataset and a single consumer-grade graphics card to assess their viability for accessible implementation. Our results demonstrate that MViT architectures achieve superior recognition accuracy of 67.98% top-1 on WLASL100 while maintaining competitive computational efficiency. Temporal perturbation analysis reveals distinct behavioural patterns across architectures: MViT models exhibit the greatest sensitivity to frame shuffling, while Swin3D shows the most robustness, suggesting varying reliance on temporal features. Error analysis further indicates that misclassifications predominantly occur between visually similar signs differing only in subtle hand configurations. These findings provide practical guidance for developing more accessible sign language recognition systems and highlight the continued need for improved temporal modelling in resource-efficient architectures.**

***Index Terms*—Sign language recognition, Fine-grained action recognition, Computer vision, Deep Learning**

I. Introduction

Many people around the world use a sign language as their primary form of communication. There are hundreds of distinct sign languages worldwide, each with unique linguistic structures and regional dialects [1, 2]. Sign languages evolved the same way as spoken languages, out of a need for people to communicate [3, 4]. In the same way as words in spoken languages are composed of sounds, signs are composed of sub-elements. These sub-elements are categorised as manual or non-manual features. Manual features are performed with the hands, where shape orientation and direction of movement affect meaning. Non-manual features are performed using the head and torso, with meaning inferred through facial expression, eye gaze and shoulder direction. A single manual or non-manual feature may distinguish a given sign [5, 6].

Most people do not know how to communicate in any sign language, and unlike some spoken languages, there is no widely available automatic translation system. Sign language translation (SLT) is a multidisciplinary field comprising Linguistics, Computer Vision and Machine Translation. In machine translation, text from one language can be translated into another; however, because sign languages use a visual modality, many have aimed to approach SLT by first transcribing sign videos into gloss[1]. This intermediary step of identifying glosses, termed sign language recognition (SLR), is analogous to converting audio to text for a spoken language [7, 8, 9, 10].

Isolated SLR involves identifying a single sign from a video, whereas the aim of continuous SLR is to identify a sequence of signs. Continuous SLR adds difficulty as it involves both localising and identifying signs in a video, and requires more computing and memory as videos are longer. Isolated SLR is still a challenging task, as the vocabulary of signs used in everyday conversation is in the thousands. This contrasts with datasets of action recognition, which typically only have a few hundred different classes [11, 12]. This aspect adds significant pressure on the scalability of video classification models.

The primary input modalities for SLR are sensor-based systems and vision-based approaches, with the latter being more scalable and accessible as they utilise ubiquitous camera technology [13, 14].

Current SLR systems are not accurate enough for practical deployment [15]. State-of-the-art isolated SLR systems achieve around 90.00% word recognition accuracy, whereas continuous SLR systems achieve about 80.00% word recognition accuracy. Practical SLT systems require SLR components to achieve greater than 99% word recognition accuracy. Certain challenging signs have been identified as responsible for the majority of false recognitions. Additionally, the best performance was achieved with data from controlled settings,

This work was undertaken in the Distributed Multimedia CoE at Rhodes University.

[1]A gloss is an identifying label assigned to a sign

while more realistic environments are even more challenging [14].

The primary contributions of this paper are as follows:

1) A controlled empirical benchmark of modern 3D CNN and vision transformer architectures for isolated SLR on the WLASL dataset, performed under consistent, resource-constrained training conditions, comparing both recognition accuracy and computational efficiency.
2) A temporal perturbation analysis through video frame shuffling, which reveals variations in reliance on spatial features over temporal sequencing in current state-of-the-art video architectures for this task.
3) A detailed error analysis that confirms the primary challenge in SLR is disambiguating visually similar signs, providing insight for future research directions.

Our findings offer practical guidance for selecting models for accessible SLR applications and highlight a critical need for improved temporal modelling in video understanding networks.

The rest of the paper is structured as follows: Section II provides an overview of related work in the field of SLR, and video classification. Following this, Section III details the model architectures and dataset, as well as the proposed fine-tuning technique and computational efficiency benchmarking. Finally, the results are provided and discussed in Section IV, along with a conclusion in Section V.

II. Related Work

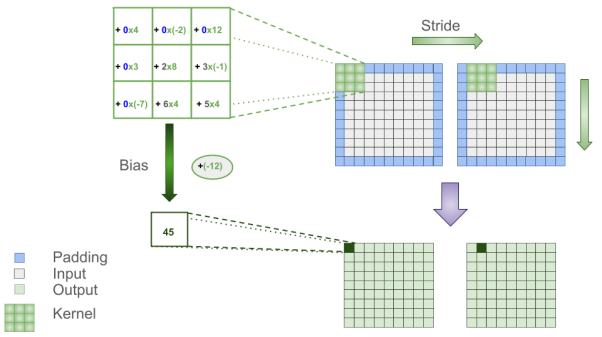
Sign language recognition requires fine-grained recognition of subtle spatio-temporal variations to distinguish different signs. Although recognition accuracy improves with higher input resolution, both spatially and temporally, these factors are constrained by GPU memory limitations [16, 17]. Pose-based approaches are a common way to reduce the dimensionality of the input by modelling the interactions of skeletal keypoints instead of raw pixels. A significant advantage of this method is that pose estimation models are pre-trained on diverse datasets, allowing them to extract normalised spatial features that may improve generalisation across signers when data is limited [18, 19]. While vision-based approaches often outperform pose-based methods in direct comparisons, the highest recognition accuracies are achieved by ensemble methods that combine modalities—such as RGB, optical flow, and skeletal data [13, 20, 21].

A. Convolutional Neural Networks

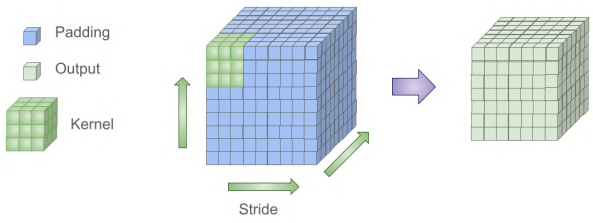
Convolutional Neural Networks (CNNs) are the foundation of many modern computer vision systems, achieving remarkable success in tasks such as image classification, object detection, and pose estimation [18, 22, 23, 24]. A convolution operation applies a set of learnable kernels (or filters) to an input feature map via a sliding dot product. The nature of this operation is defined by the kernel's dimensions:

- **1D kernels** model temporal sequences.
- **2D kernels** capture spatial features, forming the basis for image recognition.
- **3D kernels** extend this concept to spatio-temporal volumes, enabling them to learn features from both space and time simultaneously.

Fig. 1 illustrate the 2D and 3D convolution operations, in 1a and 1b respectively. Standard CNN architectures stack multiple convolutional layers, interleaved with batch normalisation and activation functions, culminating in a feed-forward classifier.



(a) 2D Convolution with kernel dimensions 3x3, and a stride of 1, over a 9x9 input with 1 layer of padding.



(b) 3D Convolution with kernel dimensions 3x3, and a stride of 1, over a 7x7 input with 1 layer of padding.

Fig. 1: A diagram of dimension preserving 2D and 3D convolution operations. Best viewed in colour.

Applying CNNs to video classification introduces the challenge of modelling time. Early approaches used 2D CNNs applied frame-by-frame, aggregating predictions through simple temporal averaging. While effective for actions defined by distinctive visuals [25], this method fails for tasks where motion dynamics are crucial, such as distinguishing between clockwise and counterclockwise rotation [26].

An alternative strategy feeds the frame-wise features from a 2D CNN into a sequence model, such as a Recurrent Neural Network (RNN) [10, 14, 27]. However, RNNs suffer from sequential processing bottlenecks and difficulty preserving long-range temporal dependencies due to vanishing gradients [12, 13, 28].

A major breakthrough came with the development of 3D CNNs, which directly learn spatio-temporal features. The inflated 3D ConvNet (I3D) was introduced as a method for

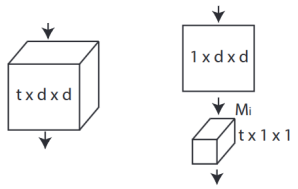


Fig. 2: On the left is a full 3D convolution with kernel size txdxd which operates over a spatio-temporal volume. On the right, the operation is split into a 2D spatial convolution, followed by a 1D temporal convolution. The number of 2D kernels (Mi) is chosen so that the (2+1)D convolutional block has the same number of parameters as the 3D convolutional block. Image taken from [29].

effectively 'inflating' an ImageNet-pre-trained Inception-v1 network into 3D [12]. This enabled the model to leverage both the powerful architectural design of the original network and the robust feature representations learned from large-scale image data. They also demonstrated that a two-stream architecture, which processes both RGB and optical flow, further enhances performance.

However, the added temporal dimension significantly increases the computational cost of 3D convolutions. This inspired the work of Tran et al. [29] on factorised convolutions. They proposed decomposing a 3D convolution into two separate operations: a 2D spatial convolution followed by a 1D temporal convolution, with ReLU activation function in between (Fig. 2). This (2+1)D factorisation not only reduces parameters but also improves accuracy and training efficiency.

Building on this, Xie et al. [30] introduced S3D, an adaptation of I3D that strategically employs factorised convolutions. A key finding was that placing 2D spatial convolutions in lower layers and 3D (or (2+1)D) spatio-temporal convolutions in higher layers improved both efficiency and accuracy.

B. Vision transformers

The Transformer architecture, introduced by Vaswani [28] for machine translation, has become the foundation for state-of-the-art models across deep learning. Its core mechanism, self-attention, enables a model to weigh the importance of all elements in a sequence when processing each element. A standard Transformer encoder layer consists of a Multi-Headed Self-Attention (MSA) block followed by a Feed-Forward Network (FFN), with residual connections and layer normalisation applied after each block. For single-headed attention, the output is a weighted sum of the values, where the weight assigned to each value is determined by the compatibility of its key with the query. A diagram of single-headed attention is shown in Fig. 3. Multi-headed attention performs this operation in parallel over multiple lower-dimensional subspaces, allowing the model to jointly attend to information from different representation subspaces.

1) From NLP to Images: The Transformer architecture was successfully adapted for image classification through the

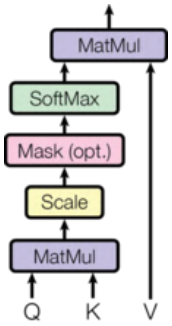


Fig. 3: Diagram of single headed attention, taken from [28]. Query, Key and Value vectors are stacked in matrices. The scaled dot product is performed between the Query and Key matrices. Part of the output may be masked to prevent the decoder from seeing future tokens. Softmax is then applied (either row-wise or column-wise) to produce the attention scores. The dot product of the attention scores with the Value matrix produces the weighted sum of the values.

Vision Transformer (ViT) [31]. The key innovation was to split an image into a sequence of fixed-size patches, treat each patch as a token, and add learnable 1D positional embeddings. This approach largely bypassed the inductive biases inherent in CNNs (e.g., translation equivariance, locality). However, by adopting the standard Transformer architecture, ViT inherited its properties of being computationally intensive and data-hungry, as well as prone to overfitting on smaller datasets. Consequently, the original ViT relied on large-scale pre-training (e.g., JFT-300M) to achieve competitive results.

The Swin Transformer [32] reintroduced useful inductive biases from CNNs to create a hierarchical vision transformer. It computes self-attention within local windows and employs a shifting window mechanism across layers to enable cross-window connections.

2) From Images to Video: Extending transformers to video introduces the formidable challenge of modelling spatio-temporal relationships without an explosion in computational complexity, as self-attention scales quadratically with input sequence length.

A primary strategy, analogous to factorised convolutions in 3D CNNs, is to factorise the attention operation across spatial and temporal dimensions [17, 33]. The Video Vision Transformer (ViViT) proposed several methods for this, such as performing spatial attention followed by temporal attention.

The Video Swin Transformer [34] adapted the hierarchical and locally-aware design of Swin to video by using 3D windows and a 3D shifting mechanism. This architecture maintains efficiency while effectively modelling local spatio-temporal features, and has proven to be a strong performer on video recognition benchmarks.

The Multiscale Vision Transformer (MViT) leverages hierarchical architectural principles established in CNNs for video processing [35, 36]. Its core innovation is a novel pooled attention mechanism: before computing attention, the

Q, K, and V tensors are pooled to reduce their spatial and/or temporal resolution. This allows MViT to efficiently model multi-scale spatio-temporal features throughout its network, significantly improving the accuracy-compute trade-off.

C. Prior Work on WLASL

The WLASL dataset [13] has become a standard benchmark for evaluating isolated sign language recognition, with most research focusing on the largest split, WLASL2000. The original benchmark was established using an I3D model [12], achieving a top-1 accuracy of 32.48%, which set a strong baseline for subsequent work.

Subsequent research has pursued higher accuracy through increasingly sophisticated architectures and multi-modal fusion. A prominent direction is pose-based methods. SignBERT represents a notable advancement in this area, using a Graph Convolutional Network (GCN) to model skeletal keypoints and a BERT-like transformer to encode temporal sequences [37]. This approach, which includes self-supervised pre-training, significantly advanced the state-of-the-art for pose-based methods, achieving 47.46% top-1 accuracy on WLASL2000. An RGB-based approach was also reported achieving 54.69%, though the specific visual backbone was not detailed.

Ensemble methods that fuse multiple modalities typically achieve the highest reported accuracies. One approach combined predictions from separate models processing RGB, optical flow, depth, and skeletal data [20]. Similarly, the Natural Language-Assisted Sign Language Recognition (NASLR) method employed a sophisticated two-stream S3D backbone that fused RGB and pose heatmaps, further incorporating language-inspired features through inter-modality mixup and label smoothing [21]. This complex approach currently represents the state-of-the-art on WLASL2000, achieving a top-1 accuracy of 61.05%.

Despite their complexity, these high-performing systems predominantly rely on 3D CNNs as their visual backbone, a consequence of both historical precedence and the former limitations of vision transformers. However, with the maturation of efficient vision transformer architectures such as MViT and Swin, their potential as a superior backbone warrants investigation. This work provides a necessary, controlled comparison of modern 3D CNNs and vision transformers under equitable, resource-constrained conditions. By evaluating these architectures in isolation on RGB data, we establish a foundational understanding of their inherent capabilities in modelling the fine-grained spatio-temporal details essential for accurate sign language recognition, and inform future architectural design choices.

III. Methodology

This study presents a systematic comparison of modern 3D CNN and vision transformer architectures for isolated sign language recognition on the WLASL dataset. To ensure a fair and controlled evaluation of the inherent capabilities of each backbone, we forgo complex architecture-specific modifications and use a consistent fine-tuning strategy across all models. We report results on the WLASL100 split and contextualise our findings against the original I3D benchmark [13].

A. Data

WLASL [13] is a large-scale isolated American Sign Language dataset collected from YouTube videos. The statistics for the full dataset are presented in Table I. In our experiments, we train and test on the WLASL100 split, which is pre-divided into a training, validation, and test set. Table II shows more specific statistics from WLASL100. There are no instances where the same signer produces the same sign in the training and testing sets. However, some signers do appear in both sets, performing different signs. Additionally, during creation, dialect labels were added to videos to ensure that all dialects in the test sets had corresponding examples in the training set.

TABLE I: WLASL Dataset Statistics.

Dataset	#Gloss	#Videos	Mean/g	#Signers
WLASL100	100	2,038	20.40	97
WLASL300	300	5,117	17.10	109
WLASL1000	1,000	13,168	13.20	116
WLASL2000	2,000	21,083	10.50	119

Mean/g: average number of videos per gloss.
Data taken from [13].

TABLE II: Statistics of WLASL100

Dataset	Split	#Ex	#Sign	Ex/g	S/g
WLASL100	Train	1442	91	14.42	11.18
	Validation	338	69	3.38	3.23
	Test	258	56	2.58	2.52

#Ex : number of examples
#Sign : number of signers
Ex/g : mean examples per gloss
S/g = mean signers per gloss.

B. Experimental Setup

Training and testing were conducted on a system with an Intel Core i5-12400 processor and 16 GiB of RAM, using a single NVIDIA GTX 1080 Ti (11 GiB VRAM). An NVIDIA RTX 3060 (12 GiB VRAM) was later acquired for computational efficiency benchmarking.

Given the computational constraints of this study, hyperparameters were chosen with strong consideration for efficient training. All models were trained with the same training hyperparameters, and the checkpoint with the lowest validation loss was used for testing. This approach has some limitations, as different architectures sometimes require distinct training regimes to perform optimally.

For all models, we used the PyTorch implementations pretrained on Kinetics 400 [12]; details of the weights and architectures are displayed in Table V. Swin3D_B had additional ImageNet-21K pretraining [38].

The final classification layer was replaced with a randomly initialised fully connected layer with dropout of 0.5 during

training. The classifier was given a higher learning rate because the weights were randomly initialised. This approach improved training times without affecting accuracy. The models were trained for a maximum of 200 epochs; however, training was terminated early if the validation loss did not decrease by at least 0.01 within 50 epochs.

An equivalent batch size of 8 was chosen because, for the number of frames and frame size provided in Table III, only a small batch size of 1-2 could fit on the GPU when using the larger models (e.g. Swin3D_B). Increasing the updates per step (batch accumulation) significantly increased training times, while offering only minimal improvements in validation accuracy.

Furthermore, because of similar memory constraints, we chose to train with 16 frames. This number is relatively small compared to many other papers, and increasing the number of frames would likely improve recognition accuracy. This is verified using smaller models in Section IV-A1.

During training and testing, a frame size of 224x224 was used, frames were uniformly sampled, and the input was normalised according to the mean and standard deviation available in the PyTorch documentation for each model. During training, the frame size was randomly cropped from the video, and a horizontal flip was applied to the video with a probability of 0.5. For testing, a centre crop of frame size was taken from the video. The training details are summarised in Table III. The training vs validation loss and accuracy curves for the best performing model (MViT_V2_S) during training are displayed in Fig. 4 and Fig. 5, respectively.

We compare our results to those reported for I3D from [13]. It should be noted that their training and testing setup differ significantly from ours. Their implementation of I3D was based on the original paper [12] in which they used a batch size of 64, 64 frames, and temporal jittering during training, as well as using all frames during testing by taking multiple views and averaging the results. Additionally, they used a dual loss function which combines prediction and localisation of a sign within a video, and was pretrained with this strategy on Kinetics. Reproducing this training strategy is outside the scope of this paper, primarily due to the need to pre-train from scratch on Kinetics using the dual loss function and the limitations of available GPU memory. Thus, the comparison aims to validate that the low-resource training strategy used remains competitive with higher-resource approaches.

Architecture efficiency benchmarking was performed separately using an NVIDIA GeForce RTX 3060 with 12 GiB VRAM. We define computational efficiency in terms of three metrics: model size (parameters), theoretical computational cost (GFLOPs), and practical performance (GPU memory usage, inference latency, and throughput). The accuracy-efficiency trade-off refers to the balance between recognition accuracy and these computational costs. The input video clips consisted of 16 frames, with a frame width and height of 224 pixels, and a batch size of 1 for each model. The batch size was set to 1 for consistency, as this was the highest batch size that could be used with this configuration for Swin3D_B. The

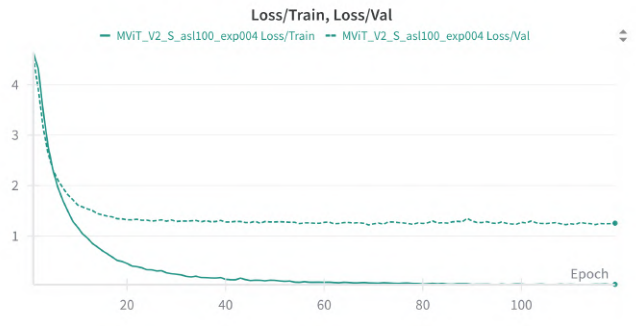


Fig. 4: Plot of the train and validation loss curves during training on the WLASL100 dataset. The graph was generated using wandb.

results were captured after warming up for 20 iterations, and then captured over 100 forward passes of the input through the model. For inference results, the models were used in PyTorch evaluation mode, with gradient tracking disabled. For the training results, the models were in training mode with gradient tracking enabled. During benchmarking and training, all models were compute-bound.

TABLE III: Training Configuration Parameters

Section	Parameter	Value
Data	Batch size	8
	No. frames	16
	Frame size	224×224
Optimizer: AdamW	Classifier LR[a]	1×10^{-3}
	Classifier WD[b]	1×10^{-7}
	Backbone LR[a]	1×10^{-5}
	Backbone WD[b]	1×10^{-4}
	Epsilon	1×10^{-3}
Scheduler: Cosine Annealing	T max	100
	Eta min	1×10^{-5}
Early stopping: Min val loss	Min delta	0.01
	Patience	50
Training	Dropout	0.50
	Max epochs	200

[a]Learning rate.
[b]Weight decay.

IV. Results

This section presents and evaluates the results from the training and efficiency benchmarking described in Section III. Table IV summarises the top-k mean per class accuracy scores on the WLASL100 test split. The different architectural families are compared in terms of their top-k average per-class recognition accuracy. The efficiency of each model in terms of memory and compute utilisation is inferred by the number of parameters and floating point operations in Table V, as well as the GPU VRAM usage, average latency and throughput of each model during training and inference in Table VI and Table VII respectively.

Vision transformer approaches achieved higher recognition accuracies, with the MViT_V2_S architecture achieving the

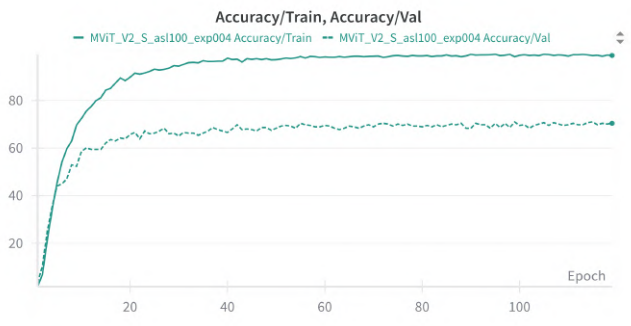


Fig. 5: Plot of the train and validation accuracy curves during training on the WLASL100 dataset. The graph was generated using wandb.

TABLE IV: Mean per class accuracy(%) achieved by each model on the WLASL100 test split.

Model	Acc@1	Acc@5	Acc@10
ResNet3D_18	45.70	75.83	86.58
ResNet(2+1)D_18	50.25	82.05	89.13
S3D	29.87	59.98	73.13
MViTV1_B	66.57	87.83	93.00
MViTV2_S	**67.98**	**89.08**	93.83
Swin3D_B^{I}	55.90	85.22	**94.58**
Swin3D_S	46.37	74.77	88.47
Swin3D_T	46.45	76.77	88.38
I3D* [13]	65.89	84.11	89.92

*Different training and testing setup.
^{I} Additional ImageNet-21K pretraining [38].

TABLE V: Pre-trained video model architectures and their performance on Kinetics-400.

Architecture	Acc@1	Acc@5	Params (M)	GFLOPs
ResNet3D_18 [29]	63.20	83.48	33.40	40.70
ResNet(2+1)_D18 [29]	67.46	86.18	31.50	40.52
S3D [30]	68.37	88.05	8.30	17.98
MViT_V1_B [35]	78.48	93.58	36.60	70.60
MViT_V2_S [36]	80.76	94.67	34.50	64.22
Swin3D_B^{r} [34]	79.40	94.40	88.00	140.67
Swin3D_B^{I} [34]	81.64	95.57	88.00	140.67
Swin3D_S [34]	79.52	94.16	49.80	82.84
Swin3D_T [34]	77.72	93.52	28.20	43.88

All models were trained and evaluated on Kinetics-400 [12].
Performance metrics obtained from PyTorch documentation.
^{I} Additional ImageNet-21K pretraining [38]
^{r} Reference model, not used in experiments

TABLE VI: Training performance metrics captured for each model in training mode over 100 forward passes, after 20 warm-up iterations, with 16 frames, frame width and height of 224 pixels, and a batch size of 1.

Model	Memory (GiB)	Latency (ms)	Throughput (videos/s)
S3D	1.61	12.03	83.12
ResNet3D_18	2.10	37.46	26.69
ResNet(2+1)D_18	4.66	69.19	14.45
Swin3D_T	3.59	43.17	23.16
Swin3D_S	5.36	73.61	13.59
Swin3D_B	6.94	104.84	9.54
MViTv2_S	4.02	51.98	19.24
MViTv1_B	3.53	40.81	24.50

TABLE VII: Inference performance metrics captured for each model in evaluation mode over 100 forward passes, after 20 warm-up iterations, with 16 frames, frame width and height of 224 pixels, and a batch size of 1.

Model	Memory (GiB)	Latency (ms)	Throughput (videos/s)
S3D	0.74	11.36	88.00
ResNet3D_18	0.95	36.35	27.51
ResNet(2+1)D_18	1.25	66.34	15.07
Swin3D_T	1.11	42.80	23.37
Swin3D_S	1.22	72.64	13.77
Swin3D_B	1.51	103.76	9.54
MViTv2_S	1.18	50.71	19.72
MViTv1_B	1.10	40.19	24.88

highest overall accuracy. Additionally, the MViT family of vision transformers were the only architecture to achieve higher accuracies than the original I3D baseline [13].

In terms of computational efficiency, S3D is the lightest model by a substantial amount, both on paper, based on the number of parameters and floating-point operations (FLOPS), and in practice, requiring the lowest memory, achieving the lowest latency, and attaining the highest throughput. Compared to the parameters and FLOPS, ResNet(2+1)D_18 had disproportionally worse performance compared to ResNet3D_18; however, this was not investigated further, and is left for future research.

The vision transformer-based models generally have a higher number of parameters and FLOPs compared to the 3D CNNs (Swin3D_T is the only one with fewer parameters than the ResNets). Swin3D_T had computational demands and inference performance similar to those of the MViT family of architectures. While Swin3D_S and Swin3D_B stood out with higher computational demand and slower inference.

In terms of the computation efficiency to accuracy trade off, the MViT family of architectures stand out for having a competitive efficieny, while achieving the highest recognition accuracies. S3D, while being very lightweight, achieved poor recognition accuracies. The Swin family of architectures, increasing in size from Swin3D_T to Swin3D_B, achieves higher accuracies, but with a steep increase in computational demands. The ResNet family of architectures have comparable efficiency to the MViT models and Swin3D_T; however, they have a lower recognition accuracy than MViT.

Comparing the results in Table IV and Table V, Swin variants and S3D, achieve relatively poor performance on WLASL compared to Kinetics, while MViT and 3D ResNets both experience a proportional drop in performance. While this performance gap may stem from insufficient temporal modelling—Kinetics features visually distinct actions while WLASL requires fine-grained motion discrimination—definitive conclusions are limited by the lack of architecture-specific hyperparameter optimisation.

A. Ablations

To further investigate the main results presented above, three key ablation tests are done.

TABLE VIII: Differences in per-class recognition accuracy(%) when increasing the number of frames from 16 to 32.

Model	Differences		
	Acc@1	Acc@5	Acc@10
ResNet3D_18	+7.73	+6.17	+2.47
ResNet(2+1)D_18	+1.07	-5.70	-4.11
S3D	**+11.35**	**+8.02**	**+5.45**

1) More frames: Many papers use higher number of frames, and research has generally shown that increasing the frame rate can significantly increase recognition accuracy [16, 17]. To determine to what extent the model's performance may be constrained by the current frame limitations, we compare recognition accuracy when increasing the number of frames from 16 to 32. This experiment was only conducted with the 3D CNNs, as they generally require less memory to train and achieved some of the lowest recognition scores. The results are presented in Table VIII. There is a significant performance increase for both S3D and ResNet3D_18. ResNet(2+1)D_18 did not receive much benefit from increasing the number of frames. It is plausible that increasing the number of frames could have also improved the performance of vision transformers. However, this investigation, along with examining why ResNet(2+1)D_18 showed no significant improvement with additional frames, falls beyond the scope of this paper and is left for future research.

2) Shuffled Videos: A phenomenon observed in coarse-grained action recognition tasks is that architectures using a 2D CNN and a simple temporal aggregation scheme, such as averaging predictions over each frame, can achieve surprisingly high prediction accuracies [25]. In contrast, fine-grained recognition tasks, such as sign language recognition, require more than strong spatial classifiers [26, 35].

To test how strongly their models relied on temporal features, Fan et al. [35] compared the inference results of their models on videos with shuffled frames from Kinetics-400 and Something-Something v2 (SSv2). They note that inference results suffered little performance degradation on Kinetics, while results on SSv2 had a substantial reduction in recognition accuracy. This was attributed to Kinetics having action classes which are visually distinct, for example: playing basketball or riding a bike, while SSv2 has action classes with very similar backgrounds and subtle variations in direction to distinguish classes, for example: unsheathing vs sheathing a sword, where both examples can be performed within the same setting.

Following this methodology, we examine recognition accuracy degradation by evaluating the models from Section III-B on frame-shuffled videos from the WLASL100 test set. The resulting differences in mean per-class accuracy, compared to normal video inference (Table IV), are summarised in Table IX.

All models experienced some level of performance degradation when tested on frame-shuffled videos. Although there were definitive trends between different architectures. The MViT and 3D ResNet variants suffered the greatest performance degradation, while the Swin family of architectures suffered the least, and S3D achieved intermediate performance degradation. This suggests that while all architectures incorporate temporal information, they do so to varying extents, with some relying more on strong 2D visual classifiers. A thorough investigation of the correlation between the architecture's ability to incorporate temporal information is left for future research.

TABLE IX: Differences in mean per class accuracy(%) between frame-shuffled videos and normal videos from WLASL100.

Model	Acc@1	Acc@5	Acc@10
ResNet3D_18	-21.55	-23.25	-22.17
ResNet(2+1)D_18	-25.27	**-35.75**	**-25.75**
S3D	-15.17	-16.62	-12.98
MViT_V1_B	-28.43	-16.25	-8.17
MViT_V2_S	**-32.68**	-17.00	-10.58
Swin3D_B	-5.92	-4.75	-4.75
Swin3D_S	-7.17	-2.58	-7.03
Swin3D_T	-7.92	-3.75	-4.38



Fig. 6: Two distinct but visually similar signs from the test set of WLASL100 [13]. The sign on the top is *Dog* and the sign on the bottom is *Thursday*.

3) Similar signs: To discern where the models' predictions are failing, we inspected which signs they struggled to predict correctly. *Thursday* was one of the signs which achieved an F1 score of 0 on the Test split, using MViT_B. When shown the sign *Thursday*, one of the predictions with the highest confidence was *Dog*. Fig. 6 depicts an example of each sign. As can be seen from the image, both are performed with a single arm, in a similar manner, rising from the bottom of the frame toward the face, making a hand gesture, then returning down and out of the frame. The only discernible difference is the shape of the hand, which also varies between different signers. Research in previous work has also demonstrated that the majority of false positives come from a handful of visually similar signs [15].

V. Conclusion

In this research, we compared a number of video classification architectures based on 3D CNNs and vision transformers on the WLASL100 dataset. The results showed that the MViT family of architectures achieved the highest recognition accuracy scores by a substantial margin, with MViTv2_S achieving

a top-1 mean per-class accuracy of 67.98%. Additionally, they have a strong trade-off between efficiency and accuracy, with comparable computational demands to those of 3D CNNs. The ResNet variants achieved the best recognition accuracies among the 3D CNNs, with ResNet(2+1)D_18 scoring a top-1 accuracy of 50.25%. S3D, while having an extremely low memory and compute footprint, achieved the worst recognition accuracies by a substantial margin (29.87% top-1). The Swin family of architectures, while achieving the second-highest recognition accuracy (Swin3D_B 55.90% top-1), had a steep increase in computational demands.

Additionally, we tested the trained models on shuffled videos to assess their dependence on temporal information and found that the MViT family of architectures experienced the greatest performance degradation, followed by the ResNets, with the Swin family being the least impacted by temporal perturbation of frames. Inspection of misclassified signs reveals that models struggle to distinguish signs that are visually very similar.

These findings reveal that while MViT architectures offer the best accuracy-efficiency trade-off for SLR, all evaluated architectures still fall short of the 99% accuracy required for practical deployment. Furthermore, while the higher recognition accuracies of vision-based transformers have outcompeted 3D CNNs for some time, they are maturing to the point where they have competitive accuracy-efficiency trade-offs. Thus, vision transformers are poised to be used as visual backbones in more advanced techniques, as their computational demands are no longer prohibitively expensive. This paves the way for increased fine-grained action understanding and enhanced recognition accuracy in sign language recognition, highlighting the need for future isolated SLR research to adopt architectures such as MViT for feature extraction.

This research has several limitations. First, hyperparameters were not individually optimised for each architecture, which may prevent models from achieving their full potential. Second, experiments were conducted on relatively small datasets, which may limit the generalisability of the results, and the comparison of the scalability of the architectures. Finally, the use of consumer-grade hardware rather than high-end data centre GPUs makes exact replication of original benchmark results challenging.

Key areas for future research include: comparing architectural scalability as the number of classes increases to more realistic sizes; exploration of multimodal approaches, particularly the integration of pose and depth information into vision transformer-based methods; architecture-specific hyperparameter optimisation; further investigation into the drop in performance of S3D and Swin achitectures relative to MViT and ResNet architectures, when transfering from the Kinetics dataset to the WLASL dataset; investigation into the unexpected behaviour of ResNet(2+1)D_18 not increasing in recognition accuracy when increasing the number of frames, and having substantially lower measured computational efficiency compared to what would be expected from literature; and investigating enhanced training strategies, including diverse augmentation techniques and multi-view inference.

References

[1] B. Woll, "Applied linguistics from the perspective of sign language and deaf studies," *Voices and Practices in Applied Linguistics: Diversifying a Discipline*, pp. 51–70, 2019.

[2] M. Moradi, D. D. Kannan, S. Asadianfam, H. Kolivand, and O. Aldhaibani, "A review of sign language systems," in *2023 16th International Conference on Developments in eSystems Engineering (DeSE)*, 2023, pp. 200–205.

[3] P. Akach, "The grammar of sign language," *Language Matters*, vol. 28, no. 1, pp. 7–35, 1997.

[4] W. Sandler and D. Lillo-Martin, "Natural sign languages," *The handbook of linguistics*, pp. 533–562, 2003.

[5] D. Brentari, *Sign language phonology*. Cambridge University Press, 2019.

[6] A. S. Elons, M. Ahmed, and H. Shedid, "Facial expressions recognition for arabic sign language translation," in *2014 9th International Conference on Computer Engineering & Systems (ICCES)*. IEEE, 2014, pp. 330–335.

[7] B. Zhou, Z. Chen, A. Clapés, J. Wan, Y. Liang, S. Escalera, Z. Lei, and D. Zhang, "Gloss-free sign language translation: Improving from visual-language pretraining," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 871–20 881.

[8] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 023–10 033.

[9] S. Sharma, R. Gupta, and A. Kumar, "Continuous sign language recognition using isolated signs data and deep transfer learning," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2023.

[10] Z. Liang, H. Li, and J. Chai, "Sign language translation: A survey of approaches and techniques," *Electronics*, vol. 12, no. 12, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/12/2678

[11] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, "The" something something" video database for learning and evaluating visual common sense," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5842–5850.

[12] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[13] D. Li, C. Rodriguez, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2020, pp. 1459–1469.

[14] N. Adaloglou, T. Chatzis, I. Papastratis, A. Stergioulas, G. T. Papadopoulos, V. Zacharopoulou, G. J. Xydopou-

los, K. Atzakas, D. Papazachariou, and P. Daras, "A comprehensive study on deep learning-based methods for sign language recognition," *IEEE transactions on multimedia*, vol. 24, pp. 1750–1762, 2021.

[15] M. Al-Qurishi, T. Khalid, and R. Souissi, "Deep learning for sign language recognition: Current techniques, benchmarks, and open issues," *IEEE Access*, vol. 9, pp. 126 917–126 951, 2021.

[16] C. Zhang, A. Gupta, and A. Zisserman, "Temporal query networks for fine-grained video understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4486–4496.

[17] A. Arnab, M. Dehghani, G. Heigold, C. Sun, D. Lu, C. Schmid, and C. Steiner, "ViViT: A video vision transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 6834–6846.

[18] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.

[19] A. Tunga, S. V. Nuthalapati, and J. Wachs, "Pose-based sign language recognition using gcn and bert," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 31–40.

[20] S. Jiang, B. Sun, L. Wang, Y. Bai, K. Li, and Y. Fu, "Skeleton aware multi-modal sign language recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3413–3423.

[21] R. Zuo, F. Wei, and B. Mak, "Natural language-assisted sign language recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 14 890–14 900.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 2002.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[25] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[26] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[27] R. Rastgoo, K. Kiani, and S. Escalera, "Sign language recognition: A deep survey," *Expert Systems with Applications*, vol. 164, p. 113794, 2021.

[28] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[29] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[30] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 305–321.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[32] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[33] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *Icml*, vol. 2, no. 3, 2021, p. 4.

[34] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3202–3211.

[35] H. Fan, B. Xiong, K. Mangalam, H. Li, F. Yan, J. Singh, and C. Feichtenhofer, "Multiscale vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 194–11 205.

[36] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer, "Mvitv2: Improved multiscale vision transformers for classification and detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4804–4814.

[37] H. Hu, W. Zhao, W. Zhou, Y. Wang, and H. Li, "Signbert: Pre-training of hand-model-aware representation for sign language recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 11 087–11 096.

[38] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," *arXiv preprint arXiv:2104.10972*, 2021.

Investigating the Effect of Copy-paste Augmentation on Fish Detection Model Generalisation

Kenneth Chieza
Department of Computer Science
Rhodes University
Makhanda, South Africa
chiezakenneth@gmail.com

Dane Brown
Department of Computer Science
Rhodes University
Makhanda, South Africa
d.brown@ru.ac.za

James Connan
Department of Computer Science
Rhodes University
Makhanda, South Africa
j.connan@ru.ac.za

Daanyaal Salie
Department of Computer Science
Rhodes University
Makhanda, South Africa
daanieson@gmail.com

***Abstract*—The use of underwater camera systems in marine science has created a need for automated fish detection models. However, the performance of these models often degrades when applied to new environments that are not seen during training. This study investigates the effectiveness of copy-paste data augmentation as a strategy to improve the generalisation of fish detection models. Experiments are conducted to test generalisation within the same geographic location, and a cross-dataset evaluation to test generalisation to a new location. Our findings reveal that the impact of copy-paste augmentation is highly context-dependent. While it can significantly degrade performance when generalising across certain environments within the same location, it substantially improves performance when generalising to a new, unseen geographic region. Furthermore, we demonstrate that pasting fish instances onto a diverse set of external, fish-free backgrounds is a particularly effective strategy, yielding the largest performance gains.**

***Index Terms*—fish detection, copy-paste, augmentation, generalisation**

I. Introduction

The growing use of underwater camera monitoring systems, such as Baited Remote Underwater Video Systems (BRUVs) and Autonomous Underwater Vehicles (AUVs), has transformed marine science by enabling large-scale, non-invasive observation of fish populations [1]. These systems generate large amounts of underwater image data that can provide valuable insights into habitat health, species abundance, and behavioural dynamics [1], [2]. However, the volume of collected data far exceeds the capacity for manual review, which is often time-consuming, labour-intensive, and prone to error. This mismatch between data availability and analysis capacity has created a pressing need for scalable and automated fish detection solutions.

Recent advances in deep learning have enabled significant progress in automated fish detection. Convolutional Neural Network-based object detection models such as You Only Look Once (YOLO) have demonstrated the potential to identify and localise fish in underwater images accurately [2]–[4]. However, these models typically require large and diverse training datasets to perform well. In practice, obtaining such datasets is challenging [5]. Variations in species, water clarity, lighting, fish orientation, and background environments can cause a model trained on one dataset to perform poorly when tested in a new setting [6], [7]. This limitation highlights a broader challenge in computer vision: achieving robust generalisation across diverse and unseen environments.

One way to address the lack of data diversity is through data augmentation [8]. Copy-paste augmentation, in particular, has gained attention as a lightweight and efficient strategy. In this approach, segmented objects are extracted from existing images and pasted onto new backgrounds, creating novel training examples without needing costly additional annotations [9]–[11]. For underwater fish detection, this method has the potential to expand datasets by introducing variations in fish placement, orientation, and context, thereby improving a model's ability to generalise.

This study investigates whether copy-paste augmentation can improve generalisation in fish detection. Specifically, we evaluate how models trained with copy-paste-augmented datasets perform when tested across different underwater environments. By analysing the impact of this augmentation technique, we aim to determine its effectiveness as a practical solution for addressing data scarcity and improving the robustness of fish detection models.

The rest of the paper is structured as follows: Section II reviews related work in data augmentation and fish detection. Section III details our proposed copy-paste pipeline. Section IV discusses the datasets, metrics used and the experimental configuration. Section V presents and discusses the results of our generalisation experiments. Finally, Section VI concludes the paper and suggests directions for future work.

II. Related Works

The copy-paste technique, first introduced by [9] as cut-paste-learn, is an object-level data augmentation method for instance detection. Unlike image-level augmentations, it offers precise control over object placement, scale, and context.

Copy-paste is typically implemented in two main ways. The first is internal pasting, where objects from the training dataset are pasted onto other images from the same dataset. This method increases object density and variability while preserving the original domain. The second, external pasting, involves pasting objects onto external background images not present in the training data, which can enhance generalisation to new environments [9], [12].

Object instances are most commonly extracted directly from the training dataset using manual annotations or segmentation models. While some methods simply crop bounding boxes, most rely on segmentation masks to isolate the object from its background [9]. Recent studies have explored using segmentation models to automate mask generation, but this can produce both high and low quality masks. Consequently, filtering procedures are often applied to discard masks that are too small, fragmented, or otherwise unsuitable [13].

Additional transformations such as scaling, rotation, and flipping are often applied to objects before pasting to introduce further variation. While many studies have shown that random placement can yield significant performance gains [10], [14], others argue that strategic, context-aware placement can improve realism and generalisation. For example, placing objects to simulate occlusion or truncation or positioning them in semantically relevant areas can enhance model robustness [11], [15]. Blending techniques to improve visual realism have shown mixed results, with some studies reporting negligible or even negative effects on performance [9], [12].

Copy-paste has been successfully applied across diverse domains, [10], [14], [16]. Many studies combine copy-paste with generative models to create a larger pool of synthetic objects or backgrounds, which has shown particular benefits for underrepresented classes [12]. Other enhancements include dynamic augmentation rates based on performance metrics [17] and strategic placement designed to simulate crowded scenes [18]. These advancements demonstrate that while simple random pasting is effective, more sophisticated approaches can further improve model performance and generalisation, particularly in challenging scenarios.

III. Methodology

The proposed copy-paste augmentation framework is illustrated in Figure 1. The pipeline is structured into three main stages: *retrieval*, *filtering*, and *pasting*. In the retrieval stage, fish instances are segmented and extracted from the dataset to create candidate cutouts. The filtering stage ensures that mostly high-quality cutouts are retained by discarding masks with segmentation errors or poor visual quality. Finally, the filtered instances are placed onto background images in the pasting stage according to the selected augmentation strategy. The following subsections describe each stage of the pipeline in detail.

A. Retrieval Stage

The retrieval stage employs a general-purpose segmentation model to extract fish cutouts from the training dataset. Unlike task-specific segmentation models, which require dedicated fish segmentation training, general-purpose models are trained to segment diverse objects across various domains, allowing plug-and-play integration into the pipeline. The model is provided with bounding box coordinates and the corresponding centre point as prompts for each fish instance, ensuring only fish objects are segmented.

The model is restricted from processing truncated or heavily occluded fish to improve segmentation quality. Truncated instances, such as fish cut off by image boundaries, typically produce incomplete masks, and heavily occluded fish often result in inaccurate segmentations. Enforcing these restrictions ensures the retrieval stage prioritises well-defined, unobstructed fish instances, reducing the likelihood of unusable cutouts. However, this restriction does not eliminate cases where fish are occluded by non-fish objects, which can still degrade segmentation quality. A subsequent filtering stage is implemented to address such remaining cases.

B. Filtering stage

An automated filtering procedure is applied to enhance the quality of the extracted masks. This process removes extracted cutouts that are too small or contain multiple disconnected components. Excessively small cutouts are often the result of segmentation errors. These cutouts are discarded by setting a size threshold. Similarly, cutouts with more than a specified number of connected components are filtered out. Multiple disconnected components can introduce inconsistencies when the objects are pasted onto new images. Such disconnections frequently arise from occlusions between fish and non-fish objects, leading to fragmented cutouts rather than coherent fish instances. Applying these filters ensures that only high-quality cutouts are retained for the subsequent pasting stage.

C. Pasting Stage

The final stage of the pipeline involves pasting segmented fish instances onto selected images. This process is controlled by an augmentation probability, which dictates whether an image is selected for copy-paste augmentation, and a pasting range, which controls the number of objects to paste onto a selected image.

Fish instances are randomly selected from the filtered mask pool and augmented through scaling and rotation to introduce instance-level variation. Cutouts are either pasted onto randomly chosen images from the same dataset they were extracted from or onto empty underwater backgrounds from a separate dataset, allowing for the generation of entirely new scenes. Candidate positions within the target image are iteratively evaluated, with placements rejected if they intersect significantly with existing bounding boxes. Once a valid

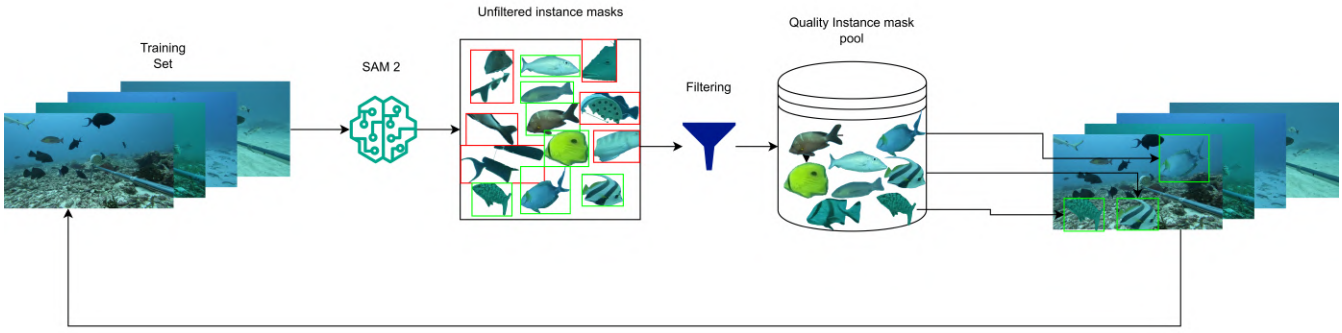


Fig. 1. Copy-Paste Pipeline

position is identified, the instance is pasted onto the image, and its bounding box label is updated accordingly.

The synthetic images generated through this stage are combined with the original training set, producing an expanded dataset enriched with additional object diversity and spatial variation. This augmented dataset is then used to train the object detection models, enhancing their ability to detect fish across varied underwater conditions. Incorporating these augmentations increases dataset diversity and provides new information that enhances the model's ability to detect fish in underwater images.

IV. Experimental Setup

A. Datasets

We conduct experiments using three distinct datasets to evaluate the effect of copy-paste augmentation on model generalisation. The first is a custom-curated dataset, Seychelles, developed in collaboration with the South African Institute for Aquatic Biodiversity (SAIAB)[1]. The second dataset is a publicly available benchmark introduced by [19], titled Annotated videos of luderick from estuaries in southeast Queensland, Australia, hereafter referred to as Seagrass. Finally, we include a small background dataset, consisting of fish-free underwater frames captured just after BRUV deployment. Both Seychelles and Seagrass contain high-resolution images collected in varied environments with different species, allowing us to test generalisation across ecological and geographic conditions, while the background dataset provides additional scene diversity for evaluating external copy-paste strategies.

The Seychelles dataset consists of frames extracted from underwater video footage captured in the Seychelles region using Baited Remote Underwater Video Systems (BRUVs). The dataset consists of various fish species, annotated with bounding boxes across varied marine environments. Table I provides the key dataset statistics, while Figure 2 illustrates the four distinct environments represented in the dataset. Each environment is characterised by unique background and habitat conditions, ensuring ecological diversity and visual variability for model training and evaluation.

[1]https://saiab.ac.za/

TABLE I
STATISTICS OF THE SEYCHELLES DATASET.

Statistic	Value
Number of images	2105
Number of null examples	92
Total number of annotations	14294
Average annotations per image	6.8

The Seagrass dataset consists of recordings of two common fish species, luderick (Girella tricuspidata) and Australian bream (Acanthopagrus australis), collected in unconstrained Seagrass habitats of southeast Queensland, Australia. The raw footage was recorded between February and July 2019 using submerged action cameras across two estuary systems: the Tweed River and Tallebudgera Creek. Following frame extraction, the data were manually annotated with instance-level labels, including bounding boxes and segmentation masks, making it suitable for object detection tasks. Dataset statistics are provided in Table II. The dataset captures substantial variability in environmental conditions such as water clarity, lighting, tides, and diverse fish orientations across 18 camera deployments. Examples of these conditions are shown in Figure 3.

TABLE II
STATISTICS OF THE SEAGRASS DATASET.

Statistic	Value
Number of images	4281
Number of null examples	0
Total number of annotations	9429
Average annotations per image	2.2

The background dataset was created by extracting frames from underwater footage captured immediately after the BRUV system was deployed but before any fish entered the field of view. All images were manually verified to ensure they were free of fish. The primary purpose of this dataset is to expose the model to a broader variety of water textures, lighting conditions, and scene compositions than those present in the internal datasets. In total, the dataset contains 11 unique background images sourced from two different locations. To increase its utility during augmentation, each background was

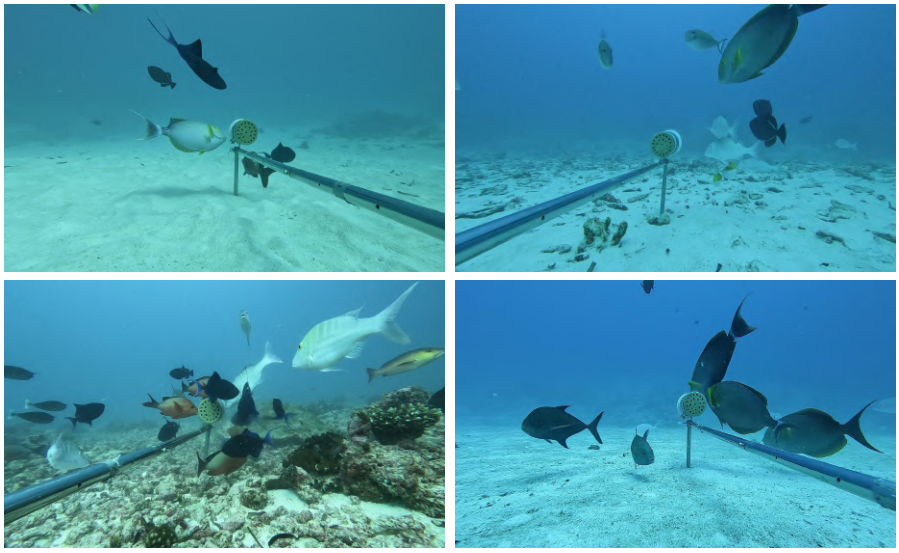


Fig. 2. Different environments of the Seychelles dataset.

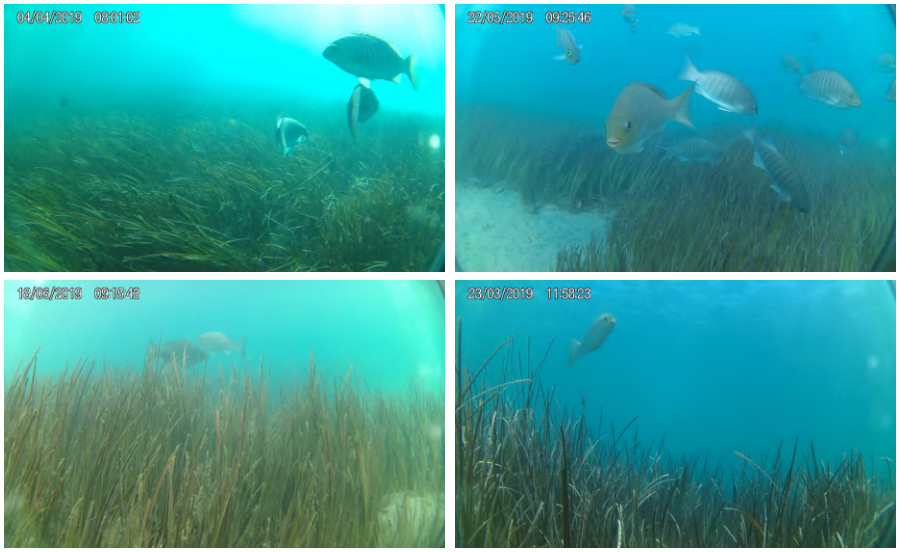


Fig. 3. Different environments of the Seagrass dataset.

duplicated 44 times, resulting in a dataset of 444 images. Example images are presented in Figure Figure 4.

B. Evaluation Metrics

Standard object detection metrics were used to evaluate the effect of copy-paste augmentation on fish detection and assess model generalisation: Precision, Recall, F1 score, and mean Average Precision at 50% IoU (mAP_{50}). When tested on unseen images, these metrics capture performance in terms of detection accuracy, localisation, and reliability.

1) Precision: Precision quantifies the proportion of predicted positives that are true positives:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

where TP denotes true positives and FP denotes false positives. A high precision indicates the model makes few incorrect detections.

2) Recall: Recall measures the proportion of actual objects correctly detected:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

where FN denotes false negatives. High recall indicates the model successfully detects most objects in the dataset.

3) F1 Score: F1 score is the harmonic mean of precision and recall, providing a balanced measure of both false positives and false negatives:

$$\text{F1} = 100 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

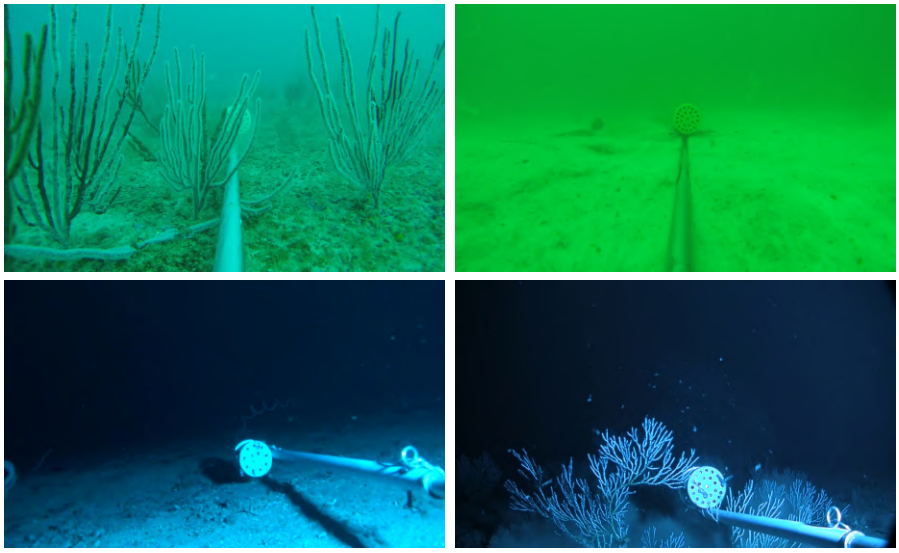


Fig. 4. Samples from the underwater background dataset.

A high F1 indicates that the model balances precision and recall well.

4) mAP_{50}: Mean Average Precision at 50% IoU (mAP_{50}) evaluates detection accuracy using the area under the precision-recall curve. A predicted bounding box is considered correct if it overlaps with the ground truth by at least 50%. Higher mAP_{50} values indicate more accurate detection and localisation, which is particularly relevant for assessing model generalisation to unseen environments.

C. Experiment Configurations

The performance of the copy-paste augmentation method was assessed using YOLOv9e object detection model implemented through the Ultralytics library [20]. The decision on using this version of YOLO was based on preliminary experiments. All experiments were conducted under consistent training conditions. Each model was trained for 300 epochs, with validation performed at the end of every epoch. Early stopping was applied with a patience of 25 epochs, halting training if validation performance did not improve over 25 consecutive epochs.

The optimiser, initial learning rate, and momentum were automatically selected. Training used a batch size of 16, input images were resized to 640 × 640 pixels, and a weight decay of 0.0005 was applied. After training, models were evaluated on the test set, with detections considered valid only if the confidence exceeded 50%. All experiments were conducted on the following hardware:

- **CPU:** Intel Core i7-13700 (13th Gen), 16 cores, 3.4 GHz base, 5.2 GHz boost
- **GPU:** NVIDIA RTX A6000, 48 GB VRAM
- **RAM:** 128 GB

When comparing copy-paste to the baseline, it is important to note that the baseline models are not trained on raw data alone but include standard data augmentations. In the copy-paste experiments, these same augmentations are applied with copy-paste, ensuring that improvements can be explicitly attributed to the copy-paste augmentation rather than the absence of other transformations. The augmentations used are listed in Table III. These were selected through a tuning process, where different augmentation configurations were tested and the combination that yielded the best performance on the baseline models was chosen. The following section describes experiments performed to evaluate the generalisation effect of copy-paste and presents their results.

TABLE III
AUGMENTATIONS USED IN TRAINING.

Augmentation	Value
Translate	0.1
Scale	0.5
Flip left-right	0.5
Mosaic	True
Erasing	0.4

V. Results and Discussion

Three experiments are performed to evaluate the effect of copy-paste augmentation on fish detection models. The first investigates how well models generalise within the same location when trained and tested across different environments. The second test generalisation across different geographic locations by training on one dataset and evaluating on another. The third explores whether pasting fish onto external background images can further improve robustness by increasing scene diversity.

The copy-paste augmentation process uses Segment Anything 2 as the general-purpose segmentation model for object extraction. Across all experiments, a consistent pasting range of 5 to 10 objects is maintained. A 50% augmentation probability is used for the first two experiments, and all

the images from the underwater backgrounds dataset undergo augmentation for the third experiment.

A. Generalisation Within the Same Location but Different Environments

This experiment evaluates the ability to generalise across environments using the Seychelles dataset by applying a leave-one-environment-out cross-validation method. The procedure for each environment, $E \in \{A, B, C, D\}$, is as follows:

1) Test Set: The test set consists of all images from environment E, which are held out from the training and validation process.
2) Training Set: Training is conducted on a combined dataset comprising images from the three environments not designated for testing.
3) A 20% stratified sample is drawn from the training pool for validation.

The experiment compares a baseline model trained without copy-paste augmentation against a model trained with copy-paste applied for each train and test set combination.

TABLE IV
LEAVE-ONE-ENVIRONMENT-OUT RESULTS ON SEYCHELLES. BEST SCORE PER TEST SET IS IN BOLD.

Test set	Train set	Aug.	Prec.	Rec.	F1	mAP50
A	B,C,D	Base	**91.10**	85.63	88.28	90.75
A	B,C,D	CP	90.64	**87.45**	**89.02**	**91.41**
B	A,C,D	Base	**93.26**	**95.40**	**94.32**	**96.80**
B	A,C,D	CP	87.61	87.05	87.33	91.79
C	A,B,D	Base	89.03	75.01	81.42	85.47
C	A,B,D	CP	**93.80**	**79.38**	**85.99**	**87.30**
D	A,B,C	Base	**95.04**	**90.66**	**92.80**	**94.66**
D	A,B,C	CP	90.89	60.04	72.32	77.00

The results in Table IV highlight that the effect of copy-paste augmentation on generalisation across environments is inconsistent. Copy-paste provided clear improvements in two cases (test environments A and C). For environment A, recall increased from 85.63 to 87.45 and mAP50 rose from 90.75 to 91.41. Similarly, for environment C, precision improved from 89.03 to 93.80, recall from 75.01 to 79.38, and mAP50 from 85.47 to 87.30. These gains suggest that copy-paste can help the model detect more fish instances and improve overall robustness when the target environment differs moderately from the training data. However, the opposite trend is observed for environments B and D. When testing on environment B, copy-paste led to a notable reduction in precision and recall, with mAP50 decreasing from 96.80 to 91.79. An even sharper drop occurred for environment D, where recall fell drastically from 90.66 to 60.04, and mAP50 decreased from 94.66 to 77.00. These results indicate that copy-paste augmentation, while beneficial in some cases, can also harm performance.

One possible explanation is that copy-paste, as implemented here, is not controlled at the species level. This can lead to imbalances in the augmented data, where particular species are over-represented in the training set. In environments where the model already performs well, such imbalances may encourage overfitting to frequently pasted species, reducing its ability to generalise to the full species distribution in the test environment.

B. Generalisation Across Different Locations

The second experiment investigates model generalisation across distinct geographic regions. Specifically, the Seychelles dataset is designated as the training source, while the Seagrass dataset serves as the independent test set. The experimental procedure is as follows:

1) Training and Validation: A training and validation split is performed exclusively on the Seychelles dataset.
2) Testing: The entire Seagrass dataset is used for testing, acting as a fully held-out test set from a different location.

We compare performance with and without copy-paste augmentation, as in the previous experiment, to provide insight into how well copy-paste augmentation enhances generalisation to new locations with differing environmental conditions and species.

TABLE V
GENERALISATION FROM SEYCHELLES (TRAINING) TO SEAGRASS (TESTING). BEST SCORE PER METRIC IS IN BOLD.

Augmentation	Prec.	Rec.	F1	mAP50
Baseline	**71.95**	15.85	25.98	35.82
Copy-paste (internal)	54.84	24.00	33.38	37.87
Copy-paste (external backgrounds)	65.54	**39.85**	**49.56**	**54.10**

The results in Table V indicate that copy-paste augmentation yields measurable improvements in recall, F1, and mAP50 when generalising to a new geographic location. Specifically, recall increased by more than 8%, and both F1 and mAP50 also improved compared to the baseline. This suggests that copy-paste augmentation enhances the model's ability to detect novel fish instances under unseen environmental conditions. A likely explanation for this improvement is that copy-paste increases the diversity of object appearances, allowing the model to learn more robust representations of fish that transfer more effectively across environments. By seeing the same objects placed in different contexts, the model may generalise better to unfamiliar scenes, even if the training backgrounds remain restricted to the Seychelles dataset. Overall performance remains low, with a recall below 25% and an F1 score below 35%. This underlines the substantial difficulty of cross-location generalisations. While copy-paste reduces some of the gap, it does not fully address the mismatch in background distribution. This limitation likely contributed to the overall low performance, as environmental differences such as water colour between Seychelles and Seagrass were not addressed. The subsequent experiment addresses this limitation.

C. Copy-Paste on External Backgrounds

The first two experiments applied copy-paste augmentation by pasting fish instances onto background images drawn

directly from the same dataset. While this approach increases instance diversity, the range of background contexts remains constrained to the environments represented in the training set. A third experiment was conducted to investigate whether a broader range of scene diversity can enhance model robustness and generalisation. This experiment involves pasting fish instances onto external background images from a separate dataset.

Fish instances from the Seychelles dataset are extracted and pasted onto images sourced from the Backgrounds dataset. The resulting synthetic images are combined with the original Seychelles dataset to form a new, expanded training set. The validation and test sets are kept unchanged, containing only original images from the Seychelles dataset.

This setup allows us to assess whether expanding the variety of background environments enhances the model's generalisation ability.

The result in Table V demonstrates that incorporating external backgrounds into the copy-paste pipeline substantially improved cross-location generalisation compared to both the baseline and when applying copy-paste by pasting onto images from the training dataset. The model trained with external backgrounds improved map50 by 18.28% compared to the base model. This model also achieved the highest recall, leading to a more balanced precision and recall tradeoff as shown by the improvement in F1 score.

These improvements suggest that introducing scene-level diversity helped the model to become less dependent on background-specific cues and more focused on learning object-level representations of fish.

VI. Conclusion

This study investigated the effectiveness of copy-paste data augmentation to improve the generalisation of fish detection models in diverse underwater environments. Our experiments revealed that the impact of this technique is context-dependent and not universally beneficial.

Our primary finding is that while copy-paste augmentation can substantially enhance model generalisation to new geographic locations, it can also be detrimental when generalising across different environments within the same location. The most significant performance gains were observed when fish instances were pasted onto a diverse set of external, fish-free backgrounds before testing on a new dataset. This approach substantially improved recall, F1-score, and mAP50, suggesting that decoupling objects from their original context is crucial for learning robust, transferable features.

The findings have practical implications for developing robust underwater monitoring systems, highlighting that the choice of augmentation strategy, particularly the source of background images, is critical. Future work should explore augmentation control based on species-level imbalances and combining copy-paste with other domain generalisation techniques.

References

[1] A. Ben Tamou, A. Benzinou, and K. Nasreddine, "Live fish species classification in underwater images by using convolutional neural networks based on incremental learning with knowledge distillation loss," *Machine Learning and Knowledge Extraction*, vol. 4, no. 3, pp. 753–767, 2022.

[2] A. B. Labao and P. C. Naval Jr, "Cascaded deep network systems with linked ensemble components for underwater fish detection in the wild," *Ecological Informatics*, vol. 52, pp. 103–121, 2019.

[3] A. Salman, S. A. Siddiqui, F. Shafait, A. Mian, M. R. Shortis, K. Khurshid, A. Ulges, and U. Schwanecke, "Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system," *ICES Journal of Marine Science*, vol. 77, no. 4, pp. 1295–1307, 2020.

[4] K. Chieza, D. Brown, J. Connan, and D. Salie, "Automated fish detection in underwater environments: Performance analysis of yolov8 and yolonas," in *Southern African Conference for Artificial Intelligence Research*. Springer, 2024, pp. 334–351.

[5] A. Saleh, I. H. Laradji, D. A. Konovalov, M. Bradley, D. Vazquez, and M. Sheaves, "A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis," *Scientific Reports*, vol. 10, no. 1, p. 14671, 2020.

[6] Z. Zhao, Y. Liu, X. Sun, J. Liu, X. Yang, and C. Zhou, "Composited fishnet: Fish detection and species recognition from low-quality underwater videos," *IEEE Transactions on Image Processing*, vol. 30, pp. 4719–4734, 2021.

[7] H. Liu, P. Song, and R. Ding, "Towards domain generalization in underwater object detection," in *2020 IEEE international conference on image processing (ICIP)*. IEEE, 2020, pp. 1971–1975.

[8] D. M. Montserrat, Q. Lin, J. Allebach, and E. J. Delp, "Training object detection and recognition cnn models using data augmentation," *Electronic Imaging*, vol. 29, pp. 27–36, 2017.

[9] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1301–1310.

[10] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2918–2928.

[11] X. Xu, B. Zhao, X. Tong, H. Xie, Y. Feng, C. Wang, C. Xiao, X. Ke, and J. Du, "A data augmentation strategy combining a modified pix2pix model and the copy-paste operator for solid waste detection with remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 8484–8491, 2022.

[12] N. Giakoumoglou, E. M. Pechlivani, and D. Tzovaras, "Generate-paste-blend-detect: Synthetic dataset for object detection in the agriculture domain," *Smart Agricultural Technology*, vol. 5, p. 100258, 2023.

[13] H. Zhao, D. Sheng, J. Bao, D. Chen, D. Chen, F. Wen, L. Yuan, C. Liu, W. Zhou, Q. Chu *et al.*, "X-paste: Revisiting scalable copy-paste for instance segmentation using clip and stablediffusion," in *International Conference on Machine Learning*. PMLR, 2023, pp. 42 098–42 109.

[14] E. Ling, D. Huang, and M. Hur, "Humans need not label more humans: Occlusion copy & paste for occluded human instance segmentation," in *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022, p. 329. [Online]. Available: https://bmvc2022.mpi-inf.mpg.de/329/

[15] Q. Guo, "Enrich the content of the image using context-aware copy paste," *CoRR*, vol. abs/2407.08151, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2407.08151

[16] Y. Jian, F. Yu, S. Singh, and D. Stamoulis, "Stable diffusion for aerial object detection," in *NeurIPS 2023 Workshop on Synthetic Data Generation with Generative AI*, 2023. [Online]. Available: https://openreview.net/forum?id=eDTdaE1yTG

[17] X. Yu, F. Li, P. Bai, Y. Liu, and Y. Chen, "Copy-paste with self-adaptation: A self-adaptive adjustment method based on copy-paste augmentation," *IET Computer Vision*, vol. 17, no. 8, pp. 936–947, 2023.

[18] J. Deng, D. Fan, X. Qiu, and F. Zhou, "Improving crowded object detection via copy-paste," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, 2023, pp. 497–505.

[19] E. M. Ditria, S. Lopez-Marcano, M. Sievers, E. L. Jinks, C. J. Brown, and R. M. Connolly, "Automating the analysis of fish abundance using object detection: optimizing animal ecology with deep learning," *Frontiers in Marine Science*, vol. 7, p. 429, 2020.

[20] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by ultralytics," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

Investigating the Effect of Copy-paste Augmentation on Fish Detection Model Generalisation

Daanyaal Salie
Department of Computer Science
Rhodes University
Makhanda, South Africa
daanieson@gmail.com

Dane Brown
Department of Computer Science
Rhodes University
Makhanda, South Africa
d.brown@ru.ac.za

James Connan
Department of Computer Science
Rhodes University
Makhanda, South Africa
j.connan@ru.ac.za

Kenneth Chieza
Department of Computer Science
Rhodes University
Makhanda, South Africa
chiezakenneth@gmail.com

***Abstract*—Manual annotation of underwater imagery is a significant bottleneck for training deep learning models for fish monitoring. Active Learning (AL) mitigates this by intelligently selecting the most informative data for annotation. This paper evaluates multiple AL strategies for fish detection using a state-of-the-art YOLOv12 model with Oriented Bounding Boxes (OBBs) on four novel datasets from diverse African aquatic ecosystems. The study systematically compares random, pure uncertainty, clustered uncertainty, and representative diversity sampling. The results show that the representative diversity strategy is superior, achieving a mean Average Precision ($mAP_{@0.5:0.95}$) of 0.713 after just five iterations. In contrast, pure uncertainty sampling performed the worst, 0.671 mAP, suffering from unstable training caused by a fixation on noisy outliers. By prioritising a balanced data distribution, diversity-based sampling builds a more robust training set, leading to superior model generalisation. These findings establish a clear and efficient pathway for developing scalable models for marine ecosystem analysis.**

***Index Terms*—Active Learning, Object Detection, Fish Detection, YOLO, Deep Learning, Computer Vision, Underwater Imagery, Uncertainty Sampling, Diversity Sampling.**

I. Introduction

Monitoring fish populations, abundance, diversity, and behaviour is fundamental to preserving marine ecosystem health and ensuring sustainable fisheries management for global food security [1]. Traditional marine research has long depended on labour-intensive methods like physical capture and diver-based visual surveys. While valuable, these approaches are often invasive, costly, and risk disturbing the ecosystems they aim to study [2].

The advent of non-intrusive underwater camera systems, such as Baited Remote Underwater Video Stations (BRUVs), has revolutionised aquatic monitoring. These technologies capture massive volumes of visual data from natural habitats with minimal disturbance [1], [3]. However, this technological leap has created a new bottleneck: the sheer quantity of data far exceeds human experts' capacity for manual analysis. This leaves researchers in a "data-rich, information-poor" state, where manual processing is slow, expensive, and prone to observer bias [4], [5]. This challenge is not unique to marine biology, with similar annotation bottlenecks appearing in diverse fields such as waste management [6] and aerial surveillance [7].

There has been a shift towards automated analysis pipelines driven by deep learning to overcome this data deluge. Convolutional Neural Networks (CNNs) have shown success in this domain, particularly object detection frameworks like YOLO (You Only Look Once), which offer real-time performance in complex visual scenes [8], [9]. However, the effectiveness of these models is constrained by two factors: the challenging underwater environment, where light attenuation and scattering degrade image quality [10], and the models' dependency on vast, manually annotated datasets for training.

This reliance on manual annotation reintroduces the human bottleneck. To break this cycle, Active Learning (AL) provides a strategic framework to reduce the manual labelling burden [11]. Instead of annotating data randomly, an AL system iteratively queries a human expert to label only the most informative samples from an unlabelled pool. This targeted approach, often guided by model uncertainty or sample diversity, can significantly accelerate model convergence and improve final performance with a fraction of the annotation effort [12].

The key contributions of this research are as follows:

- Novel African Fish Datasets: New, multi-location dataset compiled from four diverse aquatic environments, annotated with Oriented Bounding Boxes (OBBs).
- AL Framework for Fish Detection: Design and implement an AL pipeline integrating a state-of-the-art YOLOv12 model.
- Comparative Strategy Analysis: Conduct a comparison of uncertainty-based and diversity-based AL query strategies to determine their efficacy.

- Efficiency Demonstration: Empirically demonstrate that a diversity-first sampling approach significantly outperforms other methods, offering a more efficient path to robust model development.

II. Background and Related Work

A. Advancements in Real-Time Object Detection

Object detection algorithms are generally categorised into two-stage detectors (e.g., the R-CNN family), which prioritise accuracy by first proposing regions of interest and then classifying them, and one-stage detectors (e.g., YOLO), which prioritise speed by performing localisation and classification in a single pass [13]–[15]. The YOLO framework has become the predominant choice for real-time analysis applications, such as automated environmental monitoring.

The YOLO architecture has undergone rapid and significant evolution. The original model established the concept of single-pass detection, while YOLOv2 improved localisation by incorporating anchor boxes [15], [16]. YOLOv3 introduced a more powerful DarkNet-53 backbone and multi-scale detection capabilities [17]. A significant leap occurred with YOLOv4, which integrated an efficient CSPDarkNet53 backbone, a Spatial Pyramid Pooling (SPP) module, and a Path Aggregation Network (PANet) to enhance feature flow [18].

Subsequent versions continued this trend of refinement. YOLOv5 offered various model sizes for different speed-accuracy trade-offs, while YOLOv7 focused on architectural efficiencies with its Extended Efficient Layer Aggregation Network (E-ELAN) [19]. A pivotal shift came with YOLOv8, which moved to an anchor-free design, simplifying post-processing and improving flexibility [20]. Most recently, YOLOv9 introduced Programmable Gradient Information (PGI) and a Generalised Efficient Layer Aggregation Network (GELAN) to combat information loss in deep networks, further boosting parameter efficiency [21].

The pace of innovation continued with YOLOv10, which introduced a consistent dual-assignment strategy for Non-Maximum Suppression (NMS)-free training, reducing post-processing overhead [22]. YOLOv11 further enhanced feature extraction capabilities with the introduction of a C2PSA module [23]. The most recent iteration, YOLOv12, incorporates Area Attention mechanisms, allowing the model to better focus on salient object regions, which is particularly beneficial for cluttered scenes [24]. A comparative analysis of challenging African fish datasets found YOLOv12 to be the most effective architecture, delivering the highest accuracy among recent versions. Consequently, it was selected as the backbone model for this study's active learning pipeline [25].

B. The Unique Challenge of Underwater Object Detection

Despite the sophistication of modern detectors, the underwater environment presents a formidable set of challenges that degrade performance. As detailed by Pedersen et al. [10], aquatic conditions fundamentally alter image quality. Light attenuation causes colours to fade with depth, often resulting in a monochromatic blue or green hue. Light scattering from suspended particles (turbidity) blurs fine details and reduces contrast, while complex seafloor topographies and flora can create cluttered and camouflaging backgrounds. These factors collectively make it difficult for models trained on terrestrial images to generalise underwater scenes effectively.

C. Localisation Accuracy: The Case for Oriented Bounding Boxes

Beyond environmental factors, the morphology and behaviour of fish pose a critical challenge for precise localisation. Standard Horizontal Bounding Boxes (HBBs) are ill-suited for accurately encapsulating elongated objects and those that frequently appear at various angles [26]. For a diagonally swimming fish, an HBB inevitably includes large areas of background, diluting the object's features and providing misleading information to the model. In dense aggregations or shoals, overlapping HBBs can merge distinct individuals into a single detection, rendering population counts inaccurate.

To overcome this geometric imprecision, Oriented Bounding Boxes (OBBs), which rotate to fit the object's proper orientation, are essential. OBBs provide a much tighter fit, minimising the inclusion of background pixels and preserving the distinct features of each individual. A study by Salie et al. [25] has shown that transitioning from HBBs to OBBs on challenging African fish datasets boosts key precision metrics like mAP@75 by an average of 10 to 13 percentage points. This substantial improvement establishes OBBs as necessary for achieving state-of-the-art precision in underwater object detection.

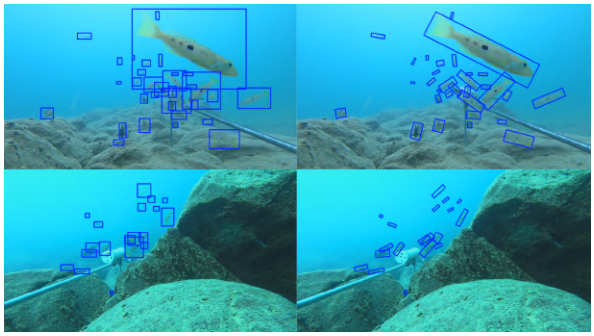


Fig. 1. Illustration of Horizontal Bounding Boxes (HBBs) vs. Oriented Bounding Boxes (OBBs). The OBB provides a tighter fit, reducing background noise and preventing the merging of individuals in dense groups [25].

D. Active Learning for Efficient Annotation

The superior performance of deep learning models, which has proven the viability of automated fish detection, is predicated on the availability of large, high-quality labelled datasets [3], [5]. While these models can achieve high accuracy, their scalability is directly hampered by the manual annotation process, a well-known bottleneck that is both time-consuming and cost-prohibitive. This challenge is not unique to marine science, with similar issues hindering progress in diverse domains from e-waste management [6] to aerial imagery analysis

[7]. To bridge the gap between the potential of automated systems and their practical, large-scale deployment, Active Learning (AL) offers a solution by directly optimising the annotation process itself.

1) Core Principles of Active Learning: As surveyed by Settles [27], the core hypothesis of AL is that a model can achieve higher performance with less data if it can strategically choose the examples from which it learns.

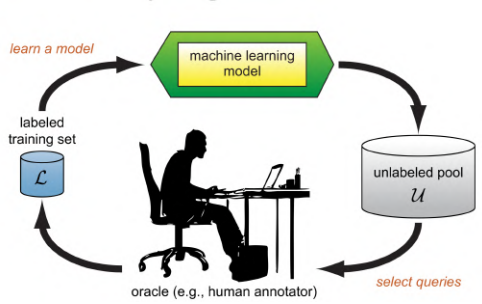


Fig. 2. The iterative cycle of an Active Learning pipeline, where the model intelligently queries a human annotator for the most informative labels [27].

2) Query Strategy Frameworks: Applying AL to object detection requires specialised strategies. As highlighted by Yu et al. [28], a good strategy must account for classification and localisation uncertainty, focus on informative local regions, and manage potential class imbalance. The two primary families of query strategies are:

Uncertainty Sampling: This is the most common AL approach, where the model requests labels for the data it is least confident about [6], [29]. Common heuristics include selecting samples with the lowest maximum softmax probability or the highest entropy in their predictive distribution. While intuitive and often effective, this method is susceptible to sampling bias. It can become fixated on outliers, near-duplicates, or noisy images that are inherently ambiguous, which may not contribute to robust generalisation [7].

Diversity Sampling: To counteract the limitations of uncertainty sampling, this strategy aims to select a batch of samples representative of the entire unlabelled data distribution. The goal is to avoid querying redundant information and ensure the model learns about all facets of the data. This is often achieved by clustering the data in a feature space and selecting samples from different clusters. Methodologies like core-set selection aim to find a small subset of points that best represent the entire dataset, thereby maximising the diversity of the queried batch.

Hybrid Approaches: Recognising the complementary nature of these two strategies, many state-of-the-art methods seek to combine them. A common approach is to create a large pool of uncertain candidates and select a diverse subset from this pool. A more sophisticated method, proposed by Yamani et al. [7], is Diverse Uncertainty Aggregation (DUA), which calculates uncertainty for each class within an image before aggregating. This encourages the selection of images containing a diverse set of uncertain objects, rather than images with many instances of a single uncertain class. Inspired by this progress, this work provides a clear, empirical comparison between uncertainty-first and diversity-first approaches in the challenging underwater domain to identify the most efficient path forward.

III. Methodology

The methodology used in this study is designed to systematically compare the effectiveness of different AL sampling strategies within a realistic fish detection pipeline.

A. The Active Learning Pipeline

The experiment follows an iterative AL process, built upon a subset of the full dataset to demonstrate the efficiency of AL. The combined dataset was partitioned using a 70/10/20 ratio for the training, validation, and test sets, respectively. The active learning process, detailed below, operated exclusively on the training split:

1) Initialisation: A small, randomly selected seed dataset of 400 images from the training split is manually annotated to train the initial YOLOv12 model (Iteration 0).
2) Feature Extraction, Prediction & Scoring: The trained model's best feature layer is used to extract embedding vectors for the entire unlabelled pool (the remainder of the training split). The model also generates predictions on these images, and each image is assigned an informativeness score based on the chosen query strategy.
3) Query Selection: The top 200 images with the most informative scores are selected for annotation.
4) Annotation & Augmentation: The selected images are annotated by the human oracle. The newly labelled data is added to the training set.
5) Retraining: The YOLOv12 model is retrained from scratch on the augmented dataset to prevent catastrophic forgetting.
6) Evaluation: The retrained model is evaluated on the fixed, held-out test and validation sets to measure its performance improvement.

This cycle is repeated for only 5 iterations due to the time limitation, adding 200 new images at each step. This means the final model is trained on only 1400 images.

B. Sampling Strategies

Four distinct strategies were evaluated in this study for scoring and selecting images. The two diversity-based methods rely on an underlying data structure provided by the HDBSCAN algorithm, which was found to be the most effective for identifying meaningful, noise-free clusters in the feature space.

- Random Sampling (Baseline): Images are selected randomly from the unlabelled pool. This serves as our baseline to measure the benefit of intelligent sampling.
- Pure Uncertainty: The uncertainty score for an image is calculated as the average of $(1-c)$ for all detected objects, where c is the model's confidence score. Images with the

highest average uncertainty are selected, without regard to diversity.

- Clustered Uncertainty: A hybrid approach that first partitions the feature space into clusters using HDBSCAN. To ensure a balanced query, a proportional number of the most uncertain samples are selected from each discovered cluster to form the final annotation batch. This method, inspired by strategies like DUA [7], ensures that queried samples are both uncertain and structurally diverse, while avoiding noisy regions of the dataset.
- Representative Diversity: After partitioning the data with HDBSCAN, this strategy selects the most representative image from each of the largest clusters (e.g., the sample with the highest membership probability score). This ensures the queried batch is a faithful, compact representation of the dataset's core visual diversity, explicitly avoiding outliers.

C. Diversity Sampling Optimisation

The effectiveness of the diversity-based sampling strategies is highly dependent on two factors: the quality of the feature vectors used to represent the data and the suitability of the clustering algorithm. Therefore, a preliminary data-driven optimisation step was incorporated into the methodology to select the best-performing components on the training set.

1) Feature Selection: To identify which layer of the YOLOv12 network produces the most discriminative features for clustering underwater scenes, an empirical evaluation was conducted by extracting feature vectors from multiple candidate layers for all images in the unlabelled pool [24]. This approach is based on the hierarchical nature of deep neural networks, where early (shallow) layers capture fundamental, low-level features such as edges and textures. As data progresses through the network, deeper layers learn to combine these simple features into more abstract and complex patterns, which are essential for identifying the objects of interest, thereby allowing us to pinpoint the most effective level of feature abstraction for the task.

2) Clustering Algorithm Selection: Two distinct clustering algorithms, K-Means and HDBSCAN [30], [31], were selected for comparison. K-Means is a computationally efficient partitioning algorithm, but it requires the number of clusters (k) to be specified beforehand. In contrast, HDBSCAN is a density-based algorithm that automatically determines the number of clusters and can identify and isolate noise points.

3) Optimisation Metric: To quantitatively assess the quality of the features and the resulting clusters, the Silhouette Score was used [31], [32]. This metric measures both cluster cohesion and separation. The score for a single sample i is given by:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{1}$$

Where $a(i)$ is the mean intra-cluster distance and $b(i)$ is the mean nearest-cluster distance for i. A score near +1 indicates dense, well-separated clusters, implying that the underlying features are highly discriminative. The combination of feature layer and clustering algorithm yielding the highest average Silhouette Score across the AL iterations would be selected for the main experiments.

D. Evaluation Metrics

To quantitatively assess model performance, a set of standard object detection metrics was employed.

1) Precision and Recall: Precision measures the model's positive predictions. It is the ratio of correctly identified fish (True Positives, TP) to the total number of positive predictions made (TP + False Positives, FP) [33]. A high score indicates that the model's positive predictions are reliable.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall (or Sensitivity) measures the model's ability to find all actual positive instances. It is the ratio of correctly identified fish (TP) to the total number of fish actually present (TP + False Negatives, FN) [33]. A high score indicates the model is comprehensive in its detections.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

2) mean Average Precision (mAP): Average Precision (AP) evaluates object detection by calculating the area under the precision-recall curve. It can be formulated as follows, where R represents recall, P represents precision, and n represents the threshold [3]:

$$AP = \sum_{n} (R_n - R_{n-1}) P_n \tag{4}$$

The mean Average Precision (mAP), as the name suggests, is the mean of AP calculated across all object classes and/or across multiple IoU thresholds [34]. It is the primary metric for evaluating the models, as it provides a single, comprehensive score that balances precision and recall across various levels of localisation accuracy. This study reports:

- mAP@0.5: AP calculated at a lenient IoU threshold of 0.50, focusing on correct detection.
- mAP@0.75: AP calculated at a stricter IoU threshold of 0.75, emphasising localisation accuracy.
- mAP@0.5:0.95: The main metric, averaged over ten IoU thresholds from 0.50 to 0.95. This gives the most holistic view of model performance across multiple IoU thresholds.

IV. Experimental Setup

A. Datasets

Four novel datasets from diverse African aquatic habitats were curated for this study in collaboration with the South African Institute for Aquatic Biodiversity, representing a significant contribution to the research community. Each dataset was annotated with OBBs for the single class, fish, using Roboflow [35]. The locations vary significantly in their environmental characteristics.

- Aldabra Atoll: A coral atoll in the Seychelles. Data was captured across multiple environments by scuba divers and proved to be the most challenging dataset due to the diversity in species, some with minimal samples, as shown by Figure 3.



Fig. 3. Example imagery from the Aldabra Atoll environment.

- Lake Tanganyika: A large rift lake in Tanzania. Data was collected using Baited Remote Underwater Video Stations (BRUVs) across four distinct environments, as shown by Figure 4.

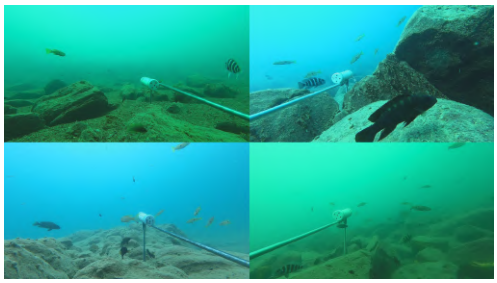


Fig. 4. Example imagery from the Lake Tanganyika environment.

- Pondoland MPA: A Marine Protected Area on South Africa's Wild Coast. This dataset was captured using BRUVs and covers five different environments, as shown in Figure 5.
- Protea Banks MPA: An offshore reef system off the coast of KwaZulu-Natal, South Africa. Imagery was captured via BRUVs across three relatively similar environments, as shown by Figure 6.

A summary of the datasets is provided in Table I.

The combined dataset of 12402 images was partitioned into a 70% training set (8681 images), a 10% validation set (1240 images), and a 20% test set (2480 images). The validation and test sets remained fixed throughout all experiments to ensure consistent evaluation.

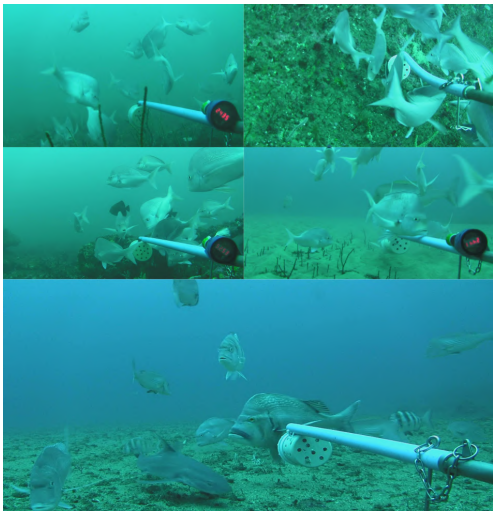


Fig. 5. Example imagery from the Pondoland MPA environment.



Fig. 6. Example imagery from the Protea Banks MPA environment.

TABLE I
OVERVIEW OF THE CURATED AFRICAN FISH DATASETS

Location	Capture Method	Enviroments	Images	Annotations
Aldabra Atoll	Scuba Diver	Varied	1910	11800
Lake Tanganyika	BRUV	4	2510	40944
Pondoland MPA	BRUV	5	4000	21344
Protea Banks MPA	BRUV	3	3,982	55306
Total			**12402**	**129354**

B. Model and Training Parameters

A consistent training procedure was used for all experiments.

- Model: YOLOv12s (small variant) with OBB support, containing approx. 9.3 million parameters and 22 layers.
- Initialisation: Transfer learning from COCO pre-trained weights [36].
- Epochs: 300 with early stopping (patience of 100).
- K-means: 5 -30 groups
- Image Size: 1024x1024 pixels.

- Hyperparameters: Learning rate of 0.01, weight decay of 0.0005, and a batch size of 16.
- Hardware: Intel i7-14700 CPU, 128GB RAM, NVIDIA RTX 6000 Ada GPU (48GB VRAM).

V. Results and Discussion

This section presents the results of our experiments, beginning with the data-driven optimisation of the pipeline components, followed by a comparison of the primary active learning strategies.

A. Pipeline Optimisation Results

The results of the preliminary experiment to optimise the diversity-based sampling components are presented here.

1) Optimal Feature Layer and Clustering Algorithm: Table II summarises the iterative evaluation of feature vectors from different network layers using two clustering algorithms. The results consistently found that features from shallow network layers, specifically layers 2 and 3, produced higher-quality clusters as measured by the Silhouette Score. This indicates that low-level features, such as edges and textures, were more informative for creating distinct groups than the complex patterns learnt in deeper layers. Notably, the clustering was primarily driven by environmental separation, grouping images with similar underwater locations and backgrounds. This demonstrates the method's ability to effectively organise large, unlabelled datasets by contextual scenes.

The HDBSCAN algorithm consistently outperformed K-Means, achieving significantly higher Silhouette Scores across almost all iterations and feature sets. This indicates that HDBSCAN produces more meaningful and well-separated clusters. This qualitative advantage is illustrated in the t-SNE visualisation in Figure 7, where HDBSCAN successfully identifies distinct data clusters while isolating noisy outlier samples, shown as grey points, also found to be the densest region from Figure 8. The analysis revealed that the outliers consisted of two main types of images: those from the Aldabra dataset, which are notable for their high environmental variation, and images from across all datasets that contained no fish. This ability to filter noise is critical for ensuring that the sampling strategies focus on representative, high-quality data.

Based on these results, the combination of features from the shallow network layers and the clustering algorithm, such as the HDBSCAN, was determined to be the best choice for structuring the feature space in the representative diversity and clustered uncertainty AL strategies.

B. Active Learning Strategy Performance

The main experimental results, summarised in Table III and in the learning curves 9, 10, 11, 12 and 13 reveal a clear performance hierarchy among the sampling strategies.

The Representative Diversity strategy was the top performer, achieving the highest final scores across nearly every key metric and culminating in a $mAP_{0.5:0.95}$ of 0.713. Its learning curve in Figure 13 shows not only the best final result but also the steepest and most stable improvement, highlighting its superior annotation efficiency. By sampling from the core of diverse visual clusters, it builds a balanced understanding of the data, which is reflected in its highest recall of 0.856 and mAP@0.75 of 0.811. This indicates that the model learns to find more fish and localise them with greater accuracy.

In contrast, Pure Uncertainty sampling was the worst-performing and most unstable strategy, ending with the lowest $mAP_{0.5:0.95}$ of 0.671. The learning curves shown confirm its erratic progress. This result confirms the pitfall of this approach: the model becomes fixated on noisy outliers. Analysis revealed that this strategy disproportionately selected images from the challenging Aldabra Atoll dataset and images with no fish across all datasets, the same subset identified as outliers by the HDBSCAN analysis. Consequently, the model wasted its limited annotation budget on ambiguous images that offered little value for generalisation. Hence, having the most gradual growth over all iterations shown by the Figures 9, 10, 11, 12 and 13.

Clustered Uncertainty, with a $mAP_{0.5:0.95}$ value of 0.709, demonstrated that balancing uncertainty with a diversity constraint successfully mitigates the failure mode of the pure approach, outperforming the random baseline. Showing improvement over the known pitfalls of pure uncertainty by introducing variability by selecting the most uncertain images over the various clusters identified. Random sampling proved to be a strong baseline with an $mAP_{0.5:0.95}$ of 0.703 and even achieved the highest precision of 0.935. This suggests random draws chose the "easier" images, leading to high confidence, but its lower recall and overall mAP confirm it builds a less comprehensive model than the diversity-guided strategies.

Ultimately, this analysis confirms the primary benefit of active learning: the ability to achieve superior model performance with substantially less labelled data. The success of the Representative Diversity and Clustered Uncertainty strategies highlights the critical importance of intelligent, diversity-guided data selection. These methods facilitated the development of robust and efficient models by focusing on a balanced set of informative examples. Conversely, the failure of the Pure Uncertainty sampling strategy, which exhausted its annotation budget on ambiguous outliers, illustrates a significant pitfall of naive uncertainty-based approaches.

Hence, the accelerated learning curves exhibited by the diversity-based methods demonstrate that the data selection methodology is of greater consequence than random sampling, enabling the development of high-performing models using only a fraction of the total available data. This data efficiency is particularly noteworthy, given that best results were achieved using a subset of only 1400 images, 16% of the available training data, highlighting the value of employing a structured data selection strategy.

VI. Conclusion

The study overall addressed the annotation bottleneck in underwater fish detection by systematically evaluating and comparing active learning strategies. The choice of sampling strategy was shown to profoundly impact annotation workload

TABLE II
SILHOUETTE SCORES FOR TOP PERFORMING LAYERS ACROSS ACTIVE LEARNING ITERATIONS AND EXPERIMENTAL SETUPS. THE HIGHEST SCORE FOR EACH ITERATION IS HIGHLIGHTED IN BOLD.

	Representative Diversity				**Clustered Uncertainty**			
	Layer 2		**Layer 3**		**Layer 2**		**Layer 3**	
Iteration	**K-Means**	**HDBSCAN**	**K-Means**	**HDBSCAN**	**K-Means**	**HDBSCAN**	**K-Means**	**HDBSCAN**
1	0.3904	0.5076	0.3646	**0.5110**	0.3904	0.5076	0.3646	**0.5110**
2	0.3790	**0.4883**	0.3619	0.4855	0.3808	0.4728	0.3579	0.4792
3	0.3438	0.4745	0.3445	0.4686	0.3517	**0.5024**	0.3620	0.4876
4	0.3606	0.4890	0.3572	0.4743	0.3747	**0.4981**	0.3664	0.4825
5	0.3806	0.4388	0.3447	0.4472	0.3729	**0.4913**	0.3616	0.4726

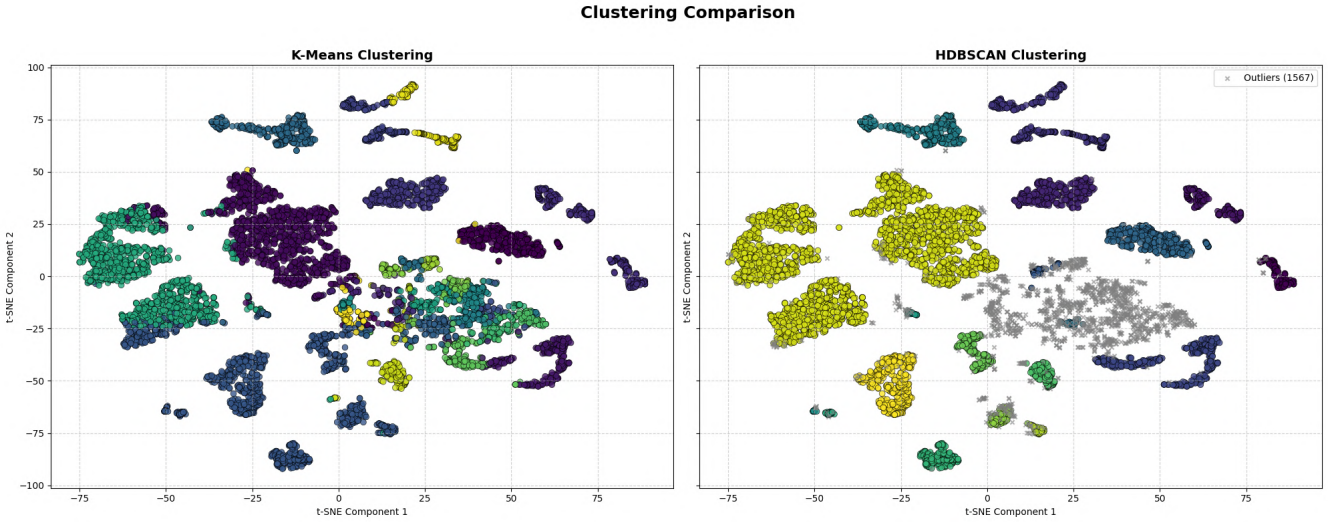


Fig. 7. A comparison of K-Means and HDBSCAN clustering for YOLOv12, showcasing the effectiveness of HDBSCAN in identifying outliers, thereby increasing the effectiveness of the clusters identified

TABLE III
PERFORMANCE OF AL SAMPLING STRATEGIES OVER 5 ITERATIONS

			Performance Metric		
Strategy	**Precision**	**Recall**	**mAP@0.5**	**mAP@0.75**	**mAP@0.5:0.95**
Iteration 0 (Initial Model)					
All	0.907	0.799	0.876	0.742	0.639
Iteration 5 (Final Model)					
Random	**0.935**	0.849	0.918	0.805	0.703
Pure Uncertainty	0.925	0.834	0.906	0.782	0.671
Clustered Uncert.	0.928	0.849	0.914	0.805	0.709
Representative Diversity	0.926	**0.856**	**0.921**	**0.811**	**0.713**

and final model performance. The key finding is that a Representative Diversity approach, which prioritises representative samples from across the data distribution, is decisively superior to both uncertainty-based methods and random sampling for this task.

The primary contributions of this work are:

1) The introduction and public release of four novel, OBB-annotated datasets from unique African aquatic environments serve as a valuable new resource for the marine research community.
2) The validation of a robust AL pipeline using the YOLOv12 architecture, confirming its suitability for challenging underwater detection tasks.
3) A rigorous comparative analysis that empirically proves diversity-based sampling is more efficient and effective than naive uncertainty-based sampling in this domain, aligning with state-of-the-art findings in other challenging computer vision tasks.
4) A clear demonstration that clustering algorithms are a powerful tool within AL to enforce sample diversity and mitigate the biases of simpler query strategies.

Future work should explore hybrid strategies that dynamically balance diversity and uncertainty, potentially incorporating performance-based weighting schemes as proposed by Yamani et al. [7] to further address class imbalance. Integrating this AL pipeline into a human-in-the-loop annotation tool for

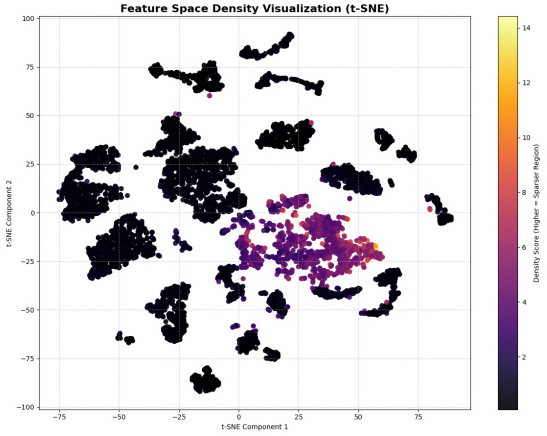


Fig. 8. t-SNE visualisation of data clustered with HDBSCAN from iteration 5 for Representative diversity. The algorithm effectively identifies distinct clusters (coloured points) while isolating noisy or outlier data points (grey points), which are less informative for diversity sampling.

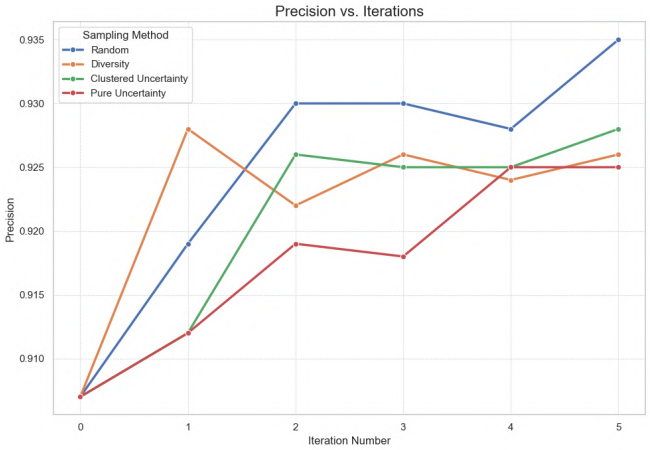


Fig. 9. Learning curve for Precision over AL iterations.

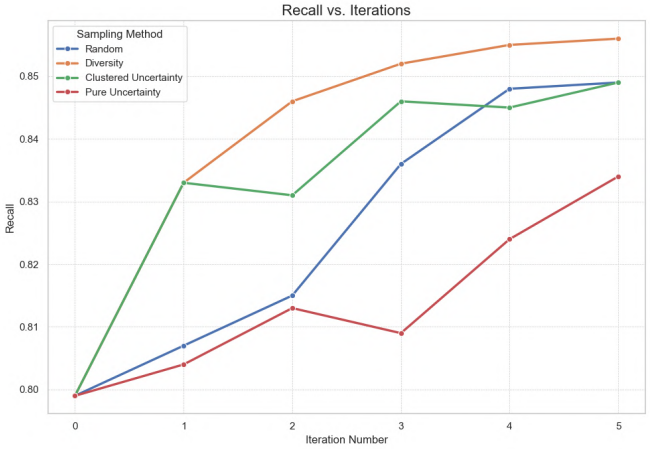


Fig. 10. Learning curve for Recall over AL iterations.

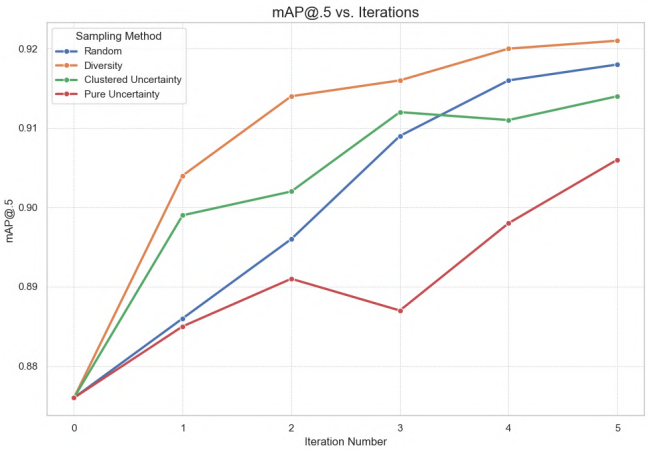


Fig. 11. Learning curve for mAP@.5 over AL iterations.

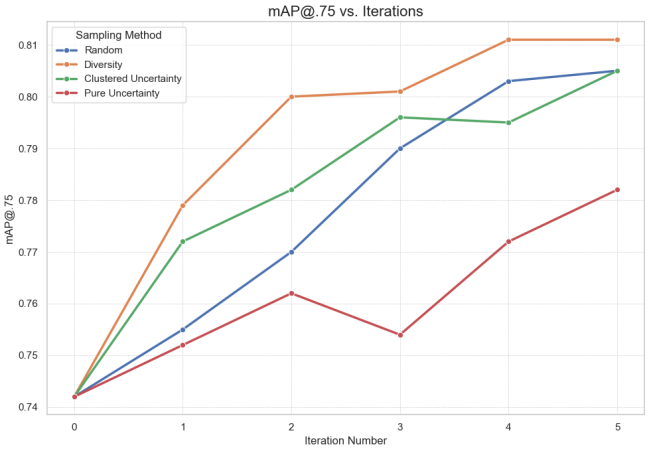


Fig. 12. Learning curve for mAP@.75 over AL iterations.

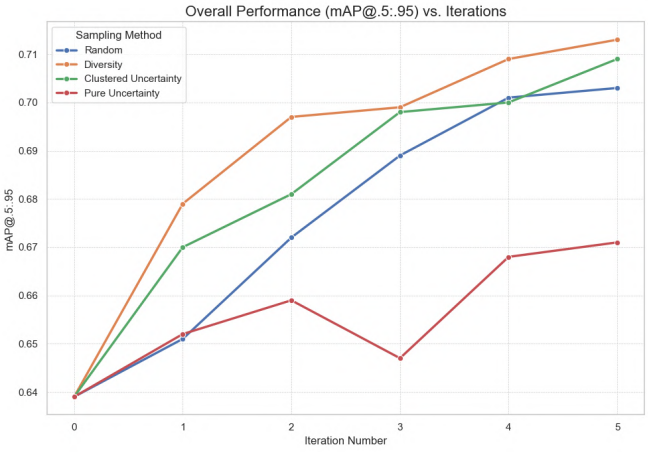


Fig. 13. Learning curve for mAP@.5:.95 over AL iterations.

marine biologists could dramatically streamline their workflow. Finally, extending this methodology from single-class detection to fine-grained, multi-class species identification

represents the next significant step toward fully automated biodiversity monitoring.

REFERENCES

[1] S. Xu, M. Zhang, W. Song, H. Mei, Q. He, and A. Liotta, "A Systematic Review and Analysis of Deep Learning-based Underwater Object Detection," *Neurocomputing*, vol. 527, pp. 204–232, 2023.

[2] P. Sarkar, S. De, and S. Gurung, "Fish Detection From Underwater Images Using YOLO and Its Challenges," in *Recent Trends in Intelligence Enabled Research*, 2022.

[3] A. A. Muksit *et al.*, "Yolo-Fish: A Robust Fish Detection Model to Detect Fish in Realistic Underwater Environment," *Ecol. Informatics*, vol. 72, 2022.

[4] K. M. Knausgård *et al.*, "Temperate Fish Detection and Classification: A Deep Learning Based Approach," *Applied Intelligence*, vol. 52, no. 6, pp. 6988–7001, 2022.

[5] S. Zhao *et al.*, "An Improved YOLO Algorithm for Fast and Accurate Underwater Object Detection," *Symmetry*, vol. 14, no. 8, 2022.

[6] Y.-J. Im, T.-W. Chang, S. Park, and H.-S. Kim, "Object Detection Models and Active Learning for Improvement of E-Waste Collection Management Systems in Korea," *Waste Management*, vol. 194, pp. 379–389, 2025.

[7] A. Yamani, A. Alyami, H. Luqman, B. Ghanem, and S. Giancola, "Active Learning for Single-stage Object Detection in UAV Images," in *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2024, Waikoloa, HI, USA, January 3-8, 2024*. IEEE, 2024, pp. 1849–1858. [Online]. Available: https://doi.org/10.1109/WACV57701.2024.00187

[8] J. H. Christensen *et al.*, "Detection, Localisation and Classification of Fish and Fish Species in Poor Conditions Using Convolutional Neural Networks," in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, 2018.

[9] K. Raza and S. Hong, "Fast and Accurate Fish Detection Design With Improved Yolo-v3 Model and Transfer Learning," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, 2020.

[10] M. Pedersen *et al.*, "Detection of Marine Animals in a New Underwater Dataset With Varying Visibility," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019*, 2019.

[11] Y.-J. Im, T.-W. Chang, S. Park, and H.-S. Kim, "Object Detection Models and Active Learning for Improvement of E-Waste Collection Management Systems in Korea," *Waste Management*, vol. 194, pp. 379–389, 2025.

[12] A. Yamani, A. Alyami, H. Luqman, B. Ghanem, and S. Giancola, "Active Learning for Single-stage Object Detection in UAV Images," in *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2024, Waikoloa, HI, USA, January 3-8, 2024*. IEEE, 2024, pp. 1849–1858. [Online]. Available: https://doi.org/10.1109/WACV57701.2024.00187

[13] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, 2014.

[14] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-Cnn: Towards Real-time Object Detection With Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[15] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, 2016.

[16] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[17] ——, "YOLOv3: An Incremental Improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[18] A. Bochkovskiy, C. Wang, and H. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *CoRR*, vol. abs/2004.10934, 2020.

[19] C. Wang, A. Bochkovskiy, and H. M. Liao, "YOLOv7: Trainable Bag-of-freebies Sets New State-of-the-art for Real-time Object Detectors," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023*, 2023.

[20] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[21] C. Wang, I. Yeh, and H. M. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," *CoRR*, vol. abs/2402.13616, 2024.

[22] A. Wang *et al.*, "YOLOv10: Real-time End-to-end Object Detection," *arXiv preprint arXiv:2405.14458*, 2024.

[23] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," *CoRR*, vol. abs/2410.17725, 2024.

[24] Y. Tian, Q. Ye, and D. S. Doermann, "YOLOv12: Attention-centric Real-time Object Detectors," *CoRR*, vol. abs/2502.12524, 2025.

[25] D. Salie, D. Brown, and K. Chieza, "Oriented Fish Detection for Enhanced Localisation," in *Proceedings of the 2025 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD 2025)*, 2025.

[26] P. Dai, S. Yao, Z. Li, S. Zhang, and X. Cao, "ACE: Anchor-free Corner Evolution for Real-time Arbitrarily-oriented Object Detection," *IEEE Trans. Image Process.*, vol. 31, pp. 4076–4089, 2022. [Online]. Available: https://doi.org/10.1109/TIP.2022.3167919

[27] B. Settles, "Active Learning Literature Survey," *Computer Sciences Technical Report 1648*, 2009.

[28] W. Yu, S. Zhu, T. Yang, and C. Chen, "Consistency-based Active Learning for Object Detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-20, 2022*. IEEE, 2022, pp. 3950–3959. [Online]. Available: https://doi.org/10.1109/CVPRW56347.2022.00440

[29] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class Active Learning for Image Classification," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, 2009, pp. 2372–2379. [Online]. Available: https://doi.org/10.1109/CVPR.2009.5206627

[30] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Hdbscan," in *Advances in Knowledge Discovery and Data Mining*. Springer Berlin Heidelberg, 2013, pp. 160–172.

[31] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation," *Electronics*, vol. 9, no. 8, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/8/1295

[32] P. J. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377042787901257

[33] N. Chen, J. Zhu, and L. Zheng, "Light-Yolo: A Study of a Lightweight YOLOv8n-based Method for Underwater Fishing Net Detection," *Applied Sciences*, vol. 14, no. 15, p. 6461, 2024. [Online]. Available: https://doi.org/10.3390/app14156461

[34] X. Li, S. Zhao, C. Chen, H. Cui, D. Li, and R. Zhao, "YOLO-FD: an Accurate Fish Disease Detection Method Based on Multi-task Learning," *Expert Syst. Appl.*, vol. 258, p. 125085, 2024. [Online]. Available: https://doi.org/10.1016/j.eswa.2024.125085

[35] B. Dwyer, J. Nelson, T. Hansen, and et al., "Roboflow," 2024, computer vision. [Online]. Available: https://roboflow.com

[36] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Computer Vision - ECCV 2014*, 2014.

Lang2Vec: language-level embeddings for robust low-resource language identification in text

Erwin Andreas Schillack and Thomas Niesler
Department of Electrical and Electronic Engineering
University of Stellenbosch, South Africa

***Abstract*—Text-based language identification remains challenging under low-resource conditions. We introduce Lang2Vec, a language-level embedding method that extends the Doc2Vec framework. In contrast to sentence-level and word-level embedding methods, Lang2Vec uses language tags to provide direct language supervision. This allows effective language separation, even in a data scarce setting. We show that using Lang2Vec embeddings as inputs to a bidirectional LSTM and also a transformer classifier leads to better language identification performance in low-resource conditions than using Word2Vec embeddings in the same way. We also show that Lang2Vec is substantially more effective at rejecting unseen languages, even when these are similar to some of the training languages. This make Lang2Vec embeddings a good candidate for application to the extraction of a target language from unconstrained multilingual input, such as web-crawled text.**

***Index Terms*—Language Identification, Low-Resource Languages, Document Embeddings, Language Embeddings, Natural Language Processing**

I. Introduction

Current language identification (LID) approaches typically rely on n-gram statistics and sentence-level representations derived from Word2Vec or similar embeddings [1], [2]. These techniques work well for high-resource languages, but they suffer in low-resource situations [3]. This is due to the absence of explicit language-level supervision, and the inability to preserve distinct boundaries between similar languages, when training data is limited.

Most African languages face these low or zero-resource challenges. Text gathered by web-crawling or similar approaches may contain a substantial amount of material in such a language of interest, but the large volume of text delivered in this way, in conjunction with the rareness of the target language, make manual dataset compilation impossible. Thus, the large volume and noisiness of short texts complicate LID in practice [4].

This paper proposes a language-level embedding (Lang2Vec) that is shown to offer superior language identification performance in such low-resource scenarios. The eventual aim is to make the automatic compilation of a text corpus in a new and under-represented language possible using this technique.

Our contributions are threefold. First, we introduce a language level embedding that injects explicit language supervision through a tag for each language during training. Second, under incremental low-resource settings, we show that these embeddings enable both long short-term memory and transformer classifiers to outperform Word2Vec inputs. Third, we demonstrate substantially improved rejection of unseen languages by lowering false-accept rates to below 1% while preserving high target recall.

The structure of the remainder of this paper is as follows. Section II reviews related work, Section III presents our Lang2Vec methodology, Section IV describes our experimental setup, Section V presents results, and Section VI concludes.

II. Background and Related Work

Modern neural networks that use word and sentence embeddings have largely replaced conventional statistical n-gram approaches in text-based language identification tasks [5]. While n-grams remain competitive, they exhibit poorer performance for short texts and text from similar languages [1], [4].

Most modern approaches to text language identification use Word2Vec embeddings [2] applied to complex neural networks, such as long short-term memory (LSTM) and transformer architectures [6]–[8]. Although these techniques perform well when datasets are large, they degrade when resources are limited.

Doc2Vec [9] extends the Word2Vec architecture by learning document-level representations and representing each document with one embedding that can be used alongside word embeddings. Although this method has proven effective in document classification tasks, it has not been considered for language identification.

Prior research on African language identification has focused on South African languages in particular, or African language families in general [3], [10]. Most languages spoken on the African continent remain unresearched, and have no digital resources such as text corpora.

Finally, the key problem posed by the unavoidable occurrence of unknown languages during the practical application of language identification systems has also received very little attention. When a language identification system is exposed to unrestricted text, such as text crawled from the internet where messages are often short and noisy [4], it is almost guaranteed to encounter languages that it did not see during training. Such languages must be reliably rejected by the system to make it useful in this situation. Most existing research has assumed a fixed and closed language set, thereby side-stepping this practical problem.

III. LANG2VEC

Lang2Vec treats all text belonging to the same language as one document. For a dataset containing K languages, each language L_i then has an associated language embedding $v_{L_i} \in \mathbb{R}^d$ that captures the specific characteristics of the language across all sentences in that language.

Lang2Vec incorporates explicit language supervision. Each sentence S_j from language L_i is associated with a language tag during training, ensuring that the language embedding is updated when any sentence from that language appears during training.

A. Rationale for Structural Features

Different languages have their own unique syntactic patterns that go beyond just the words they use, like the sentence length, punctuation use, word order and the complexity of the words. By incorporating sentence-level structural features (word count and punctuation count) alongside lexical embeddings, Lang2Vec captures these patterns that can distinguish between languages. This is especially useful in low-resource scenarios, where languages may have a large vocabulary overlap but their structural patterns are different. The projection of these features through the trainable matrix W_f allows the model to learn which structural characteristics are most informative for each language, complementing the lexical information provided by word embeddings.

B. Feature Extraction

For each sentence S_j in language L_i, the method extracts both lexical and structural features. The sentence is first tokenised into words:

$$\text{tokens}(S_j) = [w_1, w_2, \ldots, w_n] \tag{1}$$

We then compute sentence-level features capturing structural characteristics:

$$\phi(S_j) = [\text{word_count}(S_j), \text{punctuation_count}(S_j)] \tag{2}$$

Different languages exhibit varying sentence lengths and punctuation patterns. For example, Tswana sentences are typically shorter and differ in structure from English sentences.

The complete training instance for sentence S_j from language L_i is then:

$$T_{ij} = (\text{tokens}(S_j), \phi(S_j), \text{tag}(L_i)) \tag{3}$$

where $\text{tag}(L_i)$ is the language tag.

C. Distributed Memory Architecture

Lang2Vec uses the Distributed Memory (DM) architecture also employed by Doc2Vec [9] rather than the alternative Distributed Bag of Words (DBOW). The DM architecture preserves word order and context, which is useful for capturing language specific syntactic patterns.

For each language L_i we keep a trainable embedding $v_{L_i} \in \mathbb{R}^d$ stored as a column in the language embedding matrix $L \in \mathbb{R}^{d \times K}$, where K is the total number of languages the system is trained to identify. During training, $\text{tag}(L_i)$ is converted to a one-hot vector $e_{L_i} \in \mathbb{R}^K$ and retrieves $v_{L_i} = Le_{L_i}$ through matrix multiplication. The procedure is similar to how word embeddings are retrieved. The language embedding is then added to the context average and feature projection and updated by backpropagation.

The Lang2Vec model optimises both word embeddings and language embeddings by minimizing the prediction error of a target word w_t, given the surrounding context words and the language tag. The training objective is to maximise the log probability of the correct target word w_t based on the context words $w_{t-c}, \ldots, w_{t+c}$, the sentence features $\phi(S_j)$, and the language embedding v_L. Here t is the position index of the target word in the sentence. We define the context window C_t as the set of word positions surrounding the target word at position t:

$$C_t = \{\, t-c, \ldots, t-1, t+1, \ldots, t+c \,\} \cap \{1, \ldots, T\} \tag{4}$$

The structured object T_{ij}, consisting of word tokens (the previously tokenised words), sentence features, and the language tag, forms the complete input to the hidden layer. This hidden layer functions similarly to the hidden layer in the DM implementation of Doc2Vec. We define the input word embedding matrix as $W \in \mathbb{R}^{d \times V}$, where the columns are the word vectors. We then obtain the embedding $v_w \in \mathbb{R}^d$ for the word token w, which is encoded as a one hot vector $e_w \in \mathbb{R}^V$, by matrix multiplication.

$$v_w \;=\; W\, e_w \tag{5}$$

Since e_w is one-hot, this isolates one column of W. We use these embeddings to construct the context average $h_{\text{ctx}}(t)$ as shown in Equation 6.

$$h_{\text{ctx}}(t) = \frac{1}{|C_t|} \sum_{j \in C_t} v_{w_j} \tag{6}$$

Here $v_{w_j} \in \mathbb{R}^d$ is the embedding of the context token at $j \in C_t$ and $h_{\text{ctx}}(t) \in \mathbb{R}^d$.

We project the $k = 2$ dimensional sentence features, $\phi(S_j)$, into the embedding space using a trainable matrix $W_f \in \mathbb{R}^{d \times k}$ to form the projected sentence features $W_f \phi(S_j)$. These are summed with the context average $h_{\text{ctx}}(t)$ and the language embedding v_L to form the hidden representation $z(t) \in \mathbb{R}^d$.

$$z(t) \;=\; h_{\text{ctx}}(t) \;+\; v_L \;+\; W_f\, \phi(S_j) \;\in\; \mathbb{R}^d \tag{7}$$

This hidden representation $z(t)$ is passed to a linear output layer, $W' \in \mathbb{R}^{V \times d}$ to compute scores over the vocabulary, which are then normalised by a softmax to produce the probability of the target word w_t.

$$P\big(w_t \mid w_{t-c}, \ldots, w_{t+c}, v_L, \phi(S_j)\big) \;=\; \frac{e^{u_{w_t}^{\top} z(t)}}{\sum\limits_{w' \in \text{Vocab}} e^{u_{w'}^{\top} z(t)}} \tag{8}$$

In Equation 8, $u_w \in \mathbb{R}^d$ are the rows of W', the weights of the output layer. Figure 1 illustrates the Lang2Vec architecture.

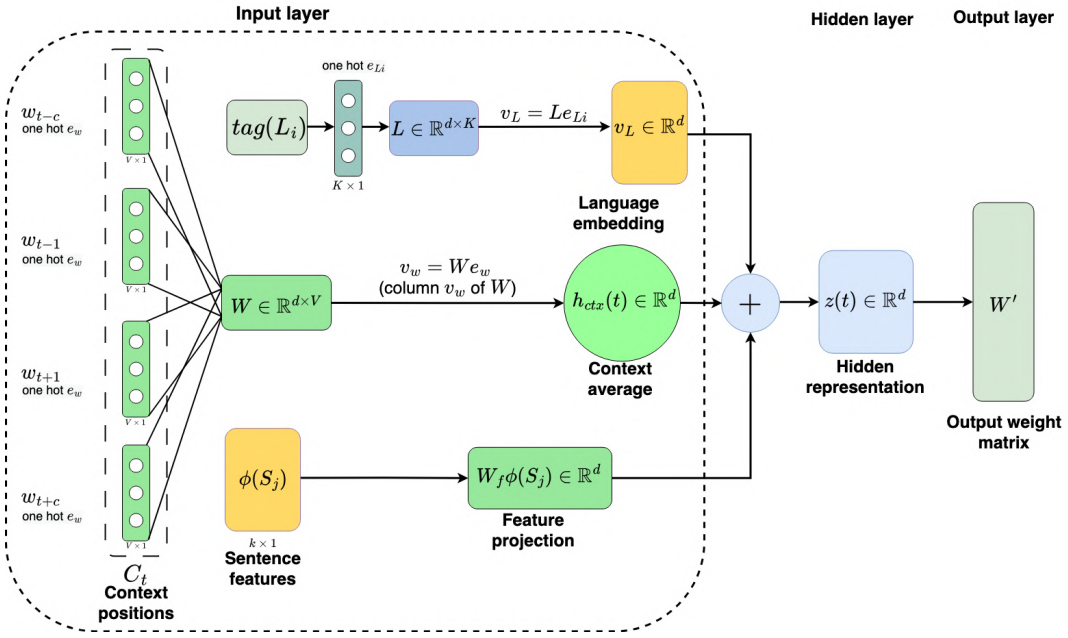


Fig. 1: A block diagram representation of the Lang2Vec architecture. The resulting hidden representation $z(t)$ is used to characterise the language of the current input sentence in a d-dimensional vector space.

D. Training Objective

We minimise the negative log-likelihood over the training set using negative sampling for computational efficiency:

$$L_t \approx -\log \sigma(u_{w_t}^T z(t)) - \sum_{n \in \mathcal{N}_t} \log \sigma(-u_n^T z(t)) \qquad (9)$$

where $\mathcal{N}_t$ represents a set of negative examples obtained from the vocabulary and σ is the sigmoid function. The following pseudocode summarizes the complete Lang2Vec training procedure:

Input: Training corpus with language tags, embedding dimension d
Initialize word embeddings W, language embeddings L, feature matrix W_f
for each training epoch **do**
  for each sentence S_j from language L_i **do**
    Extract features: $\phi(S_j) = [\text{word_count}, \text{punct_count}]$
    Retrieve language embedding: $v_L = L \cdot e_{L_i}$
    for each target word w_t in S_j **do**
      Compute context average: $h_{\text{ctx}}(t)$ from Eq. 6
      Form hidden state: $z(t) = h_{\text{ctx}}(t) + v_L + W_f\phi(S_j)$
      Predict target word via softmax: Eq. 8
      Update W, L, W_f via negative sampling
    end for
  end for
end for
Output: Trained embeddings W, L, W_f

IV. Experimental Setup

A. Datasets and Languages

We assembled a corpus combining data from the Leipzig Corpora Collection (LCC) [11], [12] and the Open Parallel Corpus (OPUS) [13]–[15] to simulate a realistic multilingual dataset [16]. This contains text in 122 different languages using the Latin alphabet. For each high-resource language, we collected 1 million sentences. For languages with fewer than 1 million sentences, we used all the available data.

We normalised all text to lowercase and limited characters to the ASCII range (hex 20-7D). Stop words were not removed, as they provide valuable linguistic information for language identification, especially for the Lang2Vec approach where the sentence features are projected into the embedding space.

We divided the data into 70% training, 20% test and 10% development sets. Hyperparameters were optimised on the development set, while 5-fold stratified cross-validation was applied during training to monitor performance. The stratified approach ensured each fold preserved the overall language distribution, preventing bias due to under representation.

Table I summarizes the key statistics of our multilingual corpus. The dataset spans 122 languages with substantial variation in resource availability, from a minimum of 7,890 sentences to a maximum of 1,000,000 sentences per language. The corpus contains 45 high-resource languages (with 1M sentences each), while 23 languages fall below 10,000 sentences, representing the low-resource scenario we aim to address.

B. Experimental Configurations

We conduct two main experiments:

TABLE I: A summary of the characteristics of the text corpus compiled for experimentation.

Statistic	Value
Total Languages	122
Total Sentences	92,210,280
Training Sentences (70%)	64,547,196
Development Sentences (10%)	9,221,028
Test Sentences (20%)	18,442,056
Min Sentences/Language	7,890
Max Sentences/Language	1,000,000
Average Sentences/Language	756,557
High-resource Languages (1M sents)	45
Languages $\leq 10k$ sentences	23
Languages $\leq 1k$ sentences	0

1) **Incremental data addition:** We systematically limit training data for target languages to the following 17 predetermined levels: 50, 100, 200, 400, 800, 1,200, ..., 300,000 and finally 700,000 sentences, the maximum available data. This simulates low-resource conditions.
2) **Unknown language detection:** We remove sets of either *similar* or *dissimilar* languages from the training set while keeping them in the test set. Three reference languages (English, German, French) remain in all training sets to provide baseline comparisons.

For the incremental data addition experiment, we selected two high-resource and two low-resource languages:

- High-resource: Afrikaans (700,000 training sentences), Bosnian (700,000 sentences).
- Low-resource: Tswana (7,182 sentences), Banjar (8,321 sentences).

To evaluate the effect of unseen languages during training, we remove ten languages per target language from the training set while leaving them in the test set. We use two complementary groups of target languages: one where the removed languages are dissimilar to the target language, and one where they are similar to the target language. Lexical similarity was estimated using the Levenshtein distance over our combined corpus. Tables II and III list the target languages and the removed languages with their respective lexical overlap in parentheses.

It is important to note that lexical similarity is used for experimental design, specifically for selecting which languages to remove from training in order to create controlled test conditions. The Lang2Vec model itself does not use lexical similarity during training or inference.

TABLE II: Target languages with removed languages that are dissimilar.

Target	Removed Languages
afr	pms, nor, tuk, lvs, lug, ltz, ina, glv, swa, srd
som	ban, bar, min, pol, lit, frr, est, fin, grn, msa
zul	gom, cos, lug, nob, mri, tuk, arg, bre, tso, bjn

TABLE III: Target languages with removed languages that are similar.

Target	Removed Languages
ita	bik, lvs, ltz, hbs, epo, ces, pam, sgs, slv, ksh
tsn	vls, bcl, bar, hsb, som, est, grn, swa, ido, xho
nor	hun, swe, vec, gom, pap, pms, bjn, pol, bos, xho

C. Model Configurations

We compare Lang2Vec with Word2Vec approaches using two neural network architectures as classifiers. In both cases hyperparameters were optimized by grid search on the development set.

- **LSTM + Word2Vec**: For each sentence, Word2Vec embeddings are processed sequentially by a bidirectional LSTM. Each word embedding has dimension $V_i^{W2V} = 50$, and the LSTM processes sequences of varying length L_i depending on sentence length. The final hidden state represents the complete sentence for classification.
- **Transformer + Word2Vec**: The transformer processes sequences of Word2Vec word embeddings in parallel using self-attention mechanisms. Word embeddings form input sequences of dimension $[L_i, V_i^{W2V}]$ where positional encoding captures word order.
- **LSTM + Lang2Vec**: Lang2Vec produces sentence-level embeddings of dimension $V_i^{L2V} = 100$ that combine word-level context, language-level characteristics, and structural features (word count, punctuation count). These complete sentence representations are processed by the LSTM architecture.
- **Transformer + Lang2Vec**: The transformer processes Lang2Vec sentence embeddings that encapsulate both local word context and global language properties within each $V_i^{L2V} = 100$ dimensional vector.

The models process batches of sentences with sequence lengths determined by the training data available for each language. The LSTM architecture uses bidirectional processing with 128 hidden dimensions, while the transformer uses 3 encoder layers with 10 attention heads and 100 hidden dimensions. All architectures were optimized on the development set using grid search over learning rates 10^{-4} to 10^{-2} and batch sizes (16-512).

D. Evaluation Metrics

We use several evaluation metrics to assess the performance of the language identification classifier, both in correctly classifying an input language and in correctly rejecting an unknown input language.

- The area under the precision-recall curve (PR-AUC) is a measure of overall classification performance.
- The recall at 99% precision indicates the recall when very few false positives are permitted. This reflects the situation in which a language corpus is distilled from web-crawled text for which the languages are unknown. The high precision ensures that the overwhelming majority of the identified sentences will be in the desired language,

and therefore the resulting corpus will contain very little text in other languages.

- The false positive rate (FPR) will be used to assess the degree to which non-target language sentences are incorrectly accepted as the target language.
- Unknown Language Rejection: Ability to correctly reject languages not seen during training. This is measured by means of confusion matrices and false accept rates.

V. Results

A. Incremental Data Addition

Table IV shows the PR-AUC performance as training data increases for the two low-resource languages, Tswana and Banjar.

TABLE IV: Area under the precision-recall curve (PR-AUC) for low-resource language identification.

Language	Configuration	Training sentences	
		$> 10^4$	10^3
Tswana	LSTM_Word2Vec	0.9101	0.2223
Tswana	LSTM_Lang2Vec	0.6406	**0.6189**
Tswana	Transformer_Word2Vec	**0.9108**	0.4954
Tswana	Transformer_Lang2Vec	0.6704	0.6519
Banjar	LSTM_Word2Vec	0.4711	0.3237
Banjar	LSTM_Lang2Vec	0.7310	0.6001
Banjar	Transformer_Word2Vec	0.5310	0.5166
Banjar	Transformer_Lang2Vec	**0.8210**	**0.6913**

The results demonstrate Lang2Vec's advantage in extremely low-resource scenarios. With only 1,000 training sentences, Lang2Vec configurations always outperform Word2Vec approaches. This pattern holds across both test languages, indicating that explicit language supervision provides stability when training data is limited.

Table V extends the analysis by showing the PR-AUC performance for the high-resource languages (Afrikaans and Bosnian) across three different data availability steps. The results reveal that Lang2Vec consistently outperforms Word2Vec methods when training data is limited (at 10^3), but this advantage falls away as more data becomes available, with the crossover occurring around 10^4 training sentences.

TABLE V: Area under the precision-recall curve (PR-AUC) for high-resource language identification.

Language	Model	Training set size (sentences)			
		10^6	10^5	10^4	10^3
Afrikaans	LR_W2V	0.8962	0.8632	0.6858	0.5501
Afrikaans	LSTM_W2V	0.9899	0.9621	0.7654	0.0312
Afrikaans	LSTM_L2V	0.9651	0.6303	0.5451	0.5278
Afrikaans	Transformer_W2V	**0.9950**	**0.9702**	**0.8201**	0.4225
Afrikaans	Transformer_L2V	0.9780	0.9201	0.6902	**0.7780**
Bosnian	LR_W2V	0.7230	0.6901	0.5102	0.4960
Bosnian	LSTM_W2V	0.7730	0.5399	0.4301	0.0260
Bosnian	LSTM_L2V	0.7432	0.5798	0.4705	0.3210
Bosnian	Transformer_W2V	**0.8920**	0.5597	0.5350	0.5185
Bosnian	Transformer_L2V	0.8530	**0.7496**	**0.6958**	**0.5950**

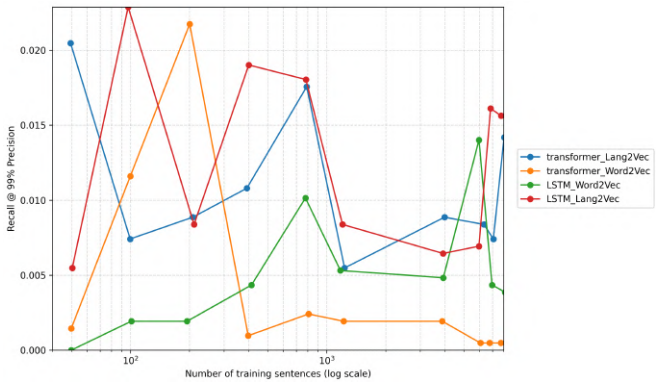
When only 1,000 sentences are available, the transformer classifier using Lang2Vec embeddings maintains reasonable performance (0.778 PR-AUC) while Word2Vec collapses dramatically (0.423 PR-AUC). This degradation is even larger when using the LSTM as classifier. We observed the same pattern for Bosnian.

This pattern reflects the differences in learning objectives. Word2Vec optimizes for semantic word relationships and requires abundant training data for stable representations, while Lang2Vec optimizes for language-level separation through supervised language tags.

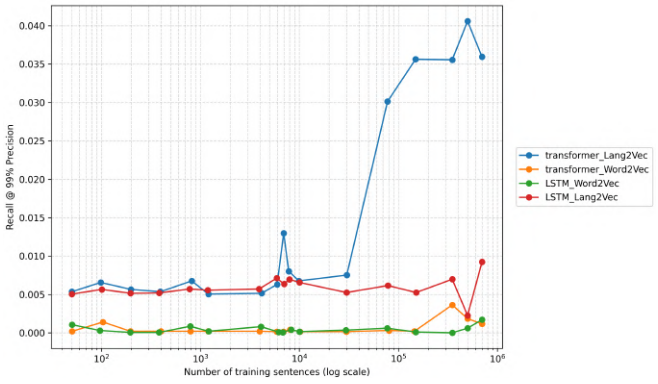
B. Recall at 99% Precision

The recall at 99% precision metric provides a strict evaluation of model performance. This metric is particularly relevant for African language evaluation where false positives can significantly impact downstream tasks.

Figure 2 compares recall at 99% precision for both Tswana and Afrikaans as training data increases. The results reveal clear patterns across resource levels that highlight the complex trade-offs between Lang2Vec and Word2Vec approaches.



(a) Tswana (low-resource)



(b) Afrikaans (high-resource)

Fig. 2: Recall at 99% precision comparison between low-resource (Tswana) and high-resource (Afrikaans) languages.

For Tswana (Figure 2a), the results shows how recall at 99% precision changes as sentences are added to the training set. When there are fewer than approximately 500 training sentences, there is no clear trend since all recall scores are very low (below 2.5%) and there are therefore only a

handful of positive classifications. However, as the number of training sentences exceeds 500, the two classifiers using Lang2Vec inputs begin to outperform the same classifiers using Word2Vec inputs in most cases. For Afrikaans (Figure 2b), more training sentences are available. We see here more clearly how the classifiers using Lang2Vec consistently outperform their Word2Vec counterparts. As the training set exceeds approximately 20,000 sentences, the transformer using Lang2Vec exhibits considerable improvements, achieving a recall of over 3.5% when all 700,000 training sentences are used. This seemingly poor performance stems from the strict evaluation constraint rather than model failure. At 99% precision, the decision threshold must be set so conservatively that almost all predictions fall below it, resulting in very few positive classifications. The classifiers using Word2Vec inputs however only show small tentative improvements by the time this upper threshold is reached.

For practical African language applications, this suggests that Lang2Vec is particularly valuable when operating under strict precision requirements, where false positives must be minimized even with limited training data.

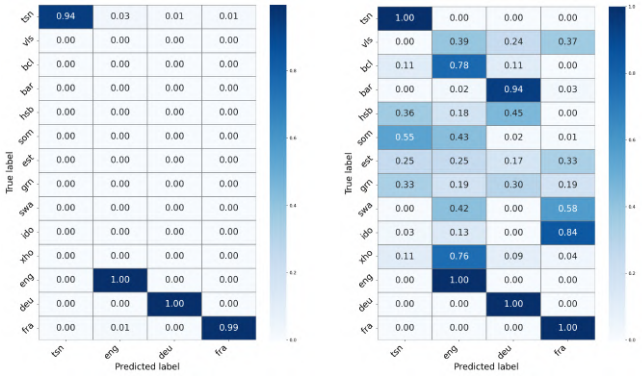
C. Unknown Language Detection

For unknown language detection, we evaluated how well the models reject languages that was unseen during training. The experimental setup used the language groupings defined in Tables II and III. The training sets always contained the target language and three reference languages (English, German, French), while 10 similar or dissimilar languages per target languages were removed from the training set but retained in the test set.

For each of the six target languages, we trained models on 113 languages (122 total minus 10 removed languages plus the target) and tested on the full test set. Results in Table VI are combined across all removed languages for each target language and combined within each similarity category. The false accept rate indicates the percentage of removed language sentences incorrectly classified as the target language (averaged across all removed languages for each target). The target recall measures the percentage of true target language sentences correctly identified. The ranges show minimum and maximum values observed across all removed languages for all the target languages combinations within each category.

Lang2Vec demonstrates competitive rejection performance both when the unknown languages are similar to some training set languages and when they are dissimilar. Figure 3 provides a direct comparison between Transformer_Lang2Vec and Transformer_Word2Vec for Tswana when ten similar languages are removed from the training set, but kept in the test set.

The Transformer_Lang2Vec achieves 94% recall on Tswana while correctly rejecting all ten removed languages (each removed language contributes $\leq 1\%$ probability to the Tswana class). In contrast, the Transformer_Word2Vec, while maintaining 100% Tswana recall, suffers many false accepts. For example, Upper Sorbian (`hsb`) is classified as Tswana in 36% of test cases, Somali (`som`) 55%, Estonian (`est`) 25%, and



(a) Transformer_Lang2Vec (b) Transformer_Word2Vec

Fig. 3: Confusion matrix when the target language is Tswana and ten similar languages have been removed from the training set but presented during testing. (a) Lang2Vec successfully rejects these similar unknown languages while (b) Word2Vec exhibits substantial false accepts.

Guarani (`grn`) 33%. The high Tswana to Somali confusion rate (55%) observed in the Word2Vec-based models, highlights a fundamental limitation of lexical embeddings when distinguishing between low-resource languages with limited training data. Analysis of the cosine similarities (Figure 4) reveals that Word2Vec embeddings place Somali sentences substantially closer to Tswana in the embedding space (similarity 0.038-0.062) than Lang2Vec embeddings (similarity $\leq$ 0.02). This geometric collapse explains the misclassification. Without explicit language-level supervision, Word2Vec relies solely on shared lexical patterns that may be similar across these languages. We observed that a similar pattern emerged for the other two target languages in Table III (`ita` and `nor`).

A similar pattern emerged when dissimilar languages were excluded, though false accept rates were generally lower (30-40% for Word2Vec vs 0.2-0.8% for Lang2Vec, see Table VI), due to greater linguistic distance from the target languages.

For languages similar to the target language, Word2Vec showed false accept rates of 50-90% across all our experiments, while Lang2Vec maintains false accept rates below 1% (Table VI). This performance difference is crucial for real world applications where unseen languages are certain to be encountered.

D. Cosine Similarity Analysis

Figure 4 illustrates the mean cosine similarity between the language embeddings of Tswana sentences, and the sentences of the removed languages. We see that the similarity between Tswana and each removed language remains low for Lang2Vec embeddings (below 0.02) for all removed languages, while they are higher for Word2Vec embeddings, particularly for Somali and Guarani.

These geometric differences explain the false accept rates observed in the confusion matrices in Figure 3. When Word2Vec embeddings place removed languages close to

TABLE VI: Language identification performance when presented with unknown languages.

Embedding	Removed	False Accept Rate	Target Recall
Word2Vec	dissimilar	32.4% (min: 28.0%, max: 41.2%)	97.2% (min: 95.0%, max: 100.0%)
Lang2Vec	dissimilar	0.40% (min: 0.11%, max: 0.81%)	95.1% (min: 92.1%, max: 98.1%)
Word2Vec	similar	68.7% (min: 52.0%, max: 89.2%)	98.1% (min: 96.1%, max: 100.0%)
Lang2Vec	similar	0.60% (min: 0.21%, max: 0.90%)	95.8% (min: 94.0%, max: 98.0%)

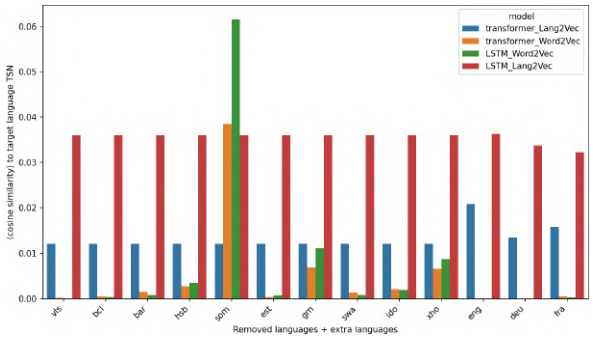


Fig. 4: Mean cosine similarity between Tswana sentence embeddings and embeddings of sentences in the removed languages. Lang2Vec maintains a clear separation from the unknown languages, while Word2Vec shows an overlap that leads to misclassification.

the target language in the embedding space, the classifier cannot reliably distinguish them. This leads to the high false acceptance rates shown in Table VI.

E. Performance Trade-offs

While Lang2Vec excels in low-resource and unknown language scenarios, Word2Vec approaches can achieve higher overall performance when enough training data is available. For Afrikaans with 700,000 training sentences, Transformer_Word2Vec achieves 0.995 PR-AUC compared to 0.978 for Lang2Vec.

Fundamental architectural differences are reflected in this trade-off. Lang2Vec is forced to learn inter-language limits during training by explicit language supervision, which prioritises language separation by design. Without this oversight, Word2Vec is only able to distinguish between languages based on lexical patterns. This requires a large amount of training data in order to create strong cross-language distinctions. However, the supervision used by Lang2Vec allows language separation even with just a few sentences.

F. Computational Considerations

Lang2Vec requires similar computational resources to Doc2Vec during training. Training time is comparable to Word2Vec when corpus sizes are similar. However, Lang2Vec requires language-tagged training data, which adds preprocessing requirements. At inference time, both approaches have similar complexity. Word2Vec requires averaging word embeddings over the sentence length, while Lang2Vec computes a single sentence embedding that incorporates both word context and structural features. In practice, inference speeds were within 10% of one another across our experiments.

VI. Conclusion

This paper presents Lang2Vec, a language-level embedding approach that demonstrates competitive performance for low-resource language identification and unknown language identification. The use of language-level supervision enables robust performance with limited training data, making it useful for African language processing applications. It was observed that Lang2Vec maintained competitive language identification performance with as few as 1,000 training sentences while Word2Vec PR-AUC degrades substantially under these conditions.

The explicit use of language labels creates strong language boundaries in the embedding space, enabling effective rejection of unknown languages with false accept rates below 1% compared to 30-90% for Word2Vec approaches. The structural characteristics captured by Lang2Vec help cross-language generalization by learning language-specific patterns that extend beyond lexical content.

While Word2Vec achieves higher performance when abundant training data is available (0.995 vs 0.978 PR-AUC for Afrikaans), Lang2Vec's language-level supervision provides a crucial advantage for practical multilingual applications where data scarcity and unknown languages are common challenges.

VII. Acknowledgements

We gratefully acknowledge support by the Department of Sports, Arts and Culture of the Republic of South Africa, and by Telkom South Africa. We would also like to thank the Stellenbosch Rhasatsha High Performance Computing facility and team for access to their facilities and technical support, which was invaluable to the successful completion of this project.

References

[1] W. B. Cavnar and J. M. Trenkle, "N-gram-based text categorization," in *Proceedings SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994, pp. 161–175.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of the 1st International Conference on Learning Representations (ICLR)*, Workshop Track, 2013.

[3] T. Jauhiainen, M. Lui, M. Zampieri, T. Baldwin, and K. Lindén, "Automatic language identification in texts: A survey," *Journal of Artificial Intelligence Research*, vol. 65, pp. 675–782, 2018.

[4] M. Z. Ansari, T. Ahmad, and A. Fatima, "Feature selection on noisy Twitter short text messages for language identification," arXiv:2007.05727 [cs.CL], 2020. [Online]. Available: https://arxiv.org/abs/2007.05727

[5] D. Jurafsky and J. H. Martin, *Speech and Language Processing* (3rd ed. draft), Online draft, 2021. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/

[6] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017, pp. 427–431.
[7] T. Kocmi and O. Bojar, "LanideNN: Multilingual language identification on character window," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017, pp. 323–327.
[8] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Word embedding and language identification using neural network models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 5, pp. 1036–1048, 2016.
[9] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014, pp. 1188–1196.
[10] A. Wright, "Automatic language identification of South African languages using n-gram language models," M.S. thesis, University of Cape Town, Cape Town, South Africa, 2007.
[11] Leipzig Corpora Collection, Univ. Leipzig. [Online]. Available: https://corpora.uni-leipzig.de/
[12] U. Quasthoff, D. Goldhahn, and T. Eckart, "Building large resources for text mining: The Leipzig Corpora Collection," in *Text Mining: From Ontology Learning to Automated Text Processing Applications*, C. Biemann and A. Mehler, Eds. Cham: Springer, 2014, pp. 3–24.
[13] J. Tiedemann, "Parallel data, tools and interfaces in OPUS," in *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, 2012, pp. 2214–2218.
[14] J. Tiedemann, "OPUS-parallel corpora for everyone," in *Proceedings of the 24th Annual Conference of the European Association for Machine Translation (EAMT) – Projects/Products*, 2016.
[15] J. Tiedemann and L. Nygaard, "The OPUS corpus – parallel and free: http://logos.uio.no/opus," in *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, European Language Resources Association (ELRA), May 2004.
[16] I. Weber, "Language identification in a highly unbalanced dataset," MEng thesis, Stellenbosch University, Stellenbosch, South Africa, 2022.

Textureless-Resilient Propagation Matching in Multi-View Stereo (TPM-MVS)

Ray Leroy Khuboni[1], Hongjun Xu [2]
Electrical, Electronic, and Computer Engineering
University of KwaZulu-Natal
Durban, South Africa, 4041
[1]KhuboniR@ukzn.ac.za, [2]Xuh@ukzn.ac.za

***Abstract*—Multi-view stereo (MVS) remains a core technique in computer vision for recovering dense 3D geometry from calibrated images. Despite substantial progress, persistent challenges in textureless regions, occlusions, thin and repetitive structures, and non-Lambertian surfaces lead to unreliable photometric consistency and insufficient view redundancy often yielding incomplete or erroneous depth maps. To address these issues, we propose textureless-resilient propagation matching (TPM-MVS), which introduces a boundary-aware checkerboard propagation sampling scheme that aggregates cost information from overlapping neighbouring regions along structural lines and adaptively shifts or extends the structured region across boundaries. We integrate this with anchor-assisted dual bilateral weighting for the photometric consistency measure to suppress irrelevant pixels and improve correspondence in textureless and repetitive areas. The design complements standard propagation and multi-hypothesis sampling, producing more complete reconstructions with negligible runtime overhead. TPM-MVS further employs multi-scale refinement to obtain reliable depth estimates from coarse to fine whilst minimizing propagation error via a confidence-guided geometric consistency term. Experimental evaluation on the public benchmark includes the ETH3D training dataset demonstrate that TPM-MVS achieves superior completeness while maintaining accuracy comparable to COLMAP, ACMH, ACMM, and representative MVSNet methods. Notably, TPM-MVS achieves an F1-score of 90.07% on the courtyard dataset under a 2 cm threshold which outperforms COLMAP's 80.49%, ACMM's 86.89%, ACMMP's 88.47%, MAR-MVS's 87.46%, and UGNet's 71.65%. The framework delivers state-of-the-art performance in challenging regions, resulting in complete and reliable 3D models.**

***Index Terms*—Multi-view stereo, PatchMatch, checkerboard propagation, boundary-aware matching, depth-map fusion, and structural region information.**

I. Introduction

Multi-view stereopsis (MVS) seeks dense correspondences across calibrated images to recover detailed 3D point clouds and textured meshes. Progress in Structure-from-Motion (SfM) [8], [34] and standardized benchmarks such as ETH3D [20], [35] has enabled rigorous evaluation, with many methods showing strong results [1]–[8], [34]. Nonetheless, MVS remains challenged by large-scale image sets, textureless or occluded regions, repetitive patterns, and non-Lambertian surfaces. Addressing these cases requires revisiting the core assumption of photometric consistency and how different MVS pipelines exploit or relax it.

Photometric consistency underpins most MVS pipelines: matching intensity patterns across calibrated views yields dense, reliable correspondences for depth estimation. However, in textureless areas and under non-Lambertian effects, cues are weak or misleading [2], [28], and repetitive patterns or occlusions further disrupt matching [6], [9]. These issues persist across the main MVS families—voxel- [11], [21]–[27], surface- [21], [24], [26], [27], patch- [1], [7], [12], [13], [28], and depth-map-based methods [2], [3], [6], [8], [10], [11], [15]–[17], [33], [34]. Patch-based approaches refine locally, while depth-map methods estimate per-view maps for later fusion. PatchMatch MVS (PM-MVS) blends both by iteratively sampling, propagating, and selecting hypotheses within a depth-map framework for efficiency and robustness [2], [7], [8], [34]; see Figure 1.

PatchMatch MVS (PM-MVS) offers efficient depth estimation by iteratively sampling hypotheses, propagating them across pixels, selecting views, and refining estimates [2], [7], [8], [12], [34], [36]. Typical propagation schemes include sequential, diffusion-like, and checkerboard updates [15], [16]. However, sequential passes introduce directional bias and edge artifacts [8], [34]; diffusion-like propagation oversmooths boundaries and is costly; and checkerboard propagation—limited to fixed eight neighbors—struggles with depth discontinuities, thin/repetitive structure, textureless regions, occlusions, and non-Lambertian effects [2].

Adaptive checkerboard propagation in ACMH/ACMM improves robustness via multi-hypothesis joint view selection and structured region sampling [15], [16], yet predefined sampling points lack connectivity, leading to discontinuities and segmentation artifacts, and reliance on reference-image bilateral weighting and joint-view inference can be unstable in ambiguous or highly non-Lambertian areas. These limitations motivate boundary-aware, structurally guided propagation as pursued in OPM-MVS [36].

To address these limitations, we introduce a boundary-aware propagation scheme that extends checkerboard sampling to mine non-local plane hypotheses [2], [7], [8], [15], [16], [34]. The method instantiates eight neighbouring structural region lines—intersecting straights around the center—and preserves four central lines (upfar, downfar, leftfar, rightfar) that expand outward. Central intersections seed locally coherent updates, while shifts along the lines gather non-local support, avoid-

ing the fixed 8-neighborhood bias of [2] and the unconstrained non-local picks of [15], [16]. This directional scaffold propagates across edges, improves continuity, captures depth discontinuities, and broadens geometric coverage—thereby enhancing depth in textureless and occluded regions.

Existing methods [2], [7], [8], [15], [16], [34] typically choose local/non-local support from the initial matching cost and apply bilateral weights from the reference view only, which can introduce artifacts when the reference is inconsistent. We address this with a dual bilateral weighted normalized cross-correlation (NCC) augmented by a re-anchoring strategy for each pixel p, we select a geometry- and texture-consistent anchor pair $(p_r^\star, p_s^\star)$ via the homography H by maximizing a score that combines bilateral color similarity in both views and geometric proximity, then center the NCC window at $(p_r^\star, p_s^\star)$ rather than (p, Hp). This symmetric anchor-guided weighting is inspired by anchor selection in UAV stitching [43], [44] to reduce bias near boundaries and under occlusion yielding a more balanced and robust photometric term that together with geometric consistency, improves correspondence in textureless regions, thin/repetitive structures, and other challenging cases.

To address these issues, Xu introduced ACMH [15] and ACMM [16] to improve propagation and densify point clouds, yet completeness in textureless regions remained limited. The extensions of ACMP [17] and ACMMP [18] leveraged planar priors to increase density, but often with visible surface noise. Other ACMM-based variants include DP-MVS [39] and MVP-Stereo [41]. Our framework extends ACMM without planar priors, it expands checkerboard sampling outward from neighboring points to harvest reliable non-local plane hypotheses and employs constrained, confidence-guided geometric consistency (Sec. III-A) to propagate robust depths. Section V-B compares our point clouds with traditional MVS and MVS-Net methods. By combining multi-scale non-local hypothesis matching, confidence-guided geometric consistency, and an effective structural sampling scheme, the approach overcomes prior PM-MVS limitations and delivers robust performance in large-scale, high-resolution reconstructions (Figure 1).

II. Overview

Given a set of input images $\mathcal{I} = \{I_i \,|\, i = 1 \cdots N\}$ with corresponding calibrated camera parameters $\mathcal{P} = \{P_i \,|\, i = 1 \cdots N\}$, the objective is to estimate depth maps $\mathcal{D} = \{D_i \,|\, i = 1 \cdots N\}$ for all input images and combine them to generate a 3D point cloud. Specifically, the depth map for a reference image I_{ref} is recovered sequentially from $\mathcal{I}$, utilizing the guidance of source images I_{src} $(\mathcal{I} - I_{\text{ref}})$.

An overview of the proposed method is depicted in Figure 1. We construct image pyramids with k scales for all input images using a downsampling factor η. The l-th scale of I_i and its associated camera parameter are represented as I_i^l and P_i^l, respectively, where $l = 0 \cdots k-1$. The raw images correspond to the finest scale of the pyramids, denoted as I_i^{k-1}. The goal is to propagate reliable depth estimates obtained from coarser scales to finer scales, ensuring minimal loss of detail in textureless regions.

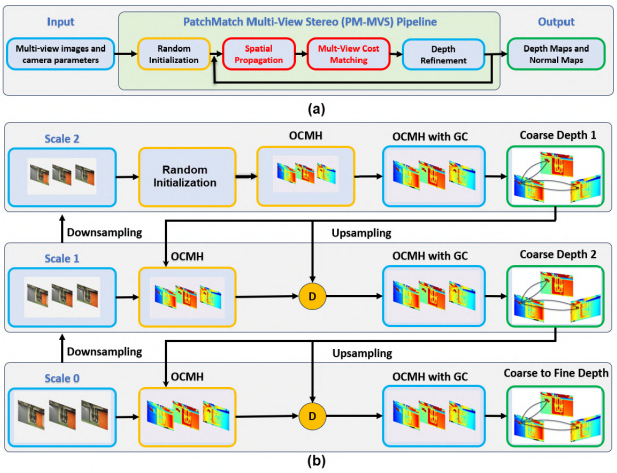


Fig. 1. (a) PM-MVS pipeline enhanced with structural sampling and Octagram Checkerboard Multi-Hypotheses (OCMH), refined using confidence-guided Geometric Consistency (GC). (b) Multi-scale PM-MVS with OCMH and GC, where depth maps are progressively refined from coarse to fine scales with photometric detail restoration (D) to ensure accurate and consistent depth estimation.

The process begins with the application of our basic MVS method with Octagram Checkerboard sampling and Multi-Hypotheses joint view selection (OCMH), which uses photometric consistency to generate initial depth maps for all images at the coarsest scale as shown in Figure 1. To improve the coherence of the depth maps, OCMH is performed with geometric consistency (GC). The depth maps are then upsampled to the next scale, propagating reliable depth estimates in texture-less regions obtained from the previous scale. To address errors arising at the coarser scale, a detail restoration step is employed. The corrected depth maps serve as the initialization for the subsequent OCMH with GC, ensuring that reliable estimates in texture-less regions are retained and optimized at the current scale. This sequence of joint bilateral upsampling [30] used in the detail restoration along with the current scale OCMH with GC is repeated until depth maps at the original image scale are obtained, similar to [15], [16]. The textureless-resilient octagram checkerboard propagation matching in a multiple scale view stereopsis is denoted as TPM-MVS throughout the paper.

A. Structured Region Information

Structured region information in Figure 2, implies that pixels within a relatively large area can approximately be represented by the same 3D plane. Building on this concept, our basic MVS method, which incorporates Octagram Checkerboard sampling and Multi-Hypotheses Joint View Selection (OCMH), is designed to sample more reliable candidate hypotheses for propagation and select views with higher credibility for multi-view matching cost aggregation in Figure 2. The specifics of OCMH are outlined as follows.

B. Random Initialization

Following the approach outlined in [2], we start by randomly generating a hypothesis consisting of depth and normal values to define a 3D plane for each pixel in the reference image I_{ref}. The matching cost for each hypothesis is computed using plane-induced homography across the $N-1$ source images [29]. Specifically, the camera parameters of the reference image I_{ref} are represented as $P_{\text{ref}} = K_{\text{ref}}\left[R_{\text{ref}} \,|\, t_{\text{ref}}\right]$, and the camera parameters of the j-th source image are given as $P_j = K_j\left[R_j \,|\, t_j\right]$. The homography relating them is computed as in [29]:

$$H_j = K_j\left(R_j - \frac{t_j \boldsymbol{n}^\top}{d}\right) K_{\text{ref}}^{-1}. \tag{1}$$

Using this transformation, the corresponding patch in the j-th source image is identified for the patch of pixel p in I_{ref}. The matching cost is then calculated using the dual bilateral weighted adaptation of normalized cross-correlation which is similar to [8], [15], [16], [19], [34]. Finally, the top K matching costs are averaged to derive the initial multi-view matching cost, which serves as the input for subsequent propagation.

C. Boundary Aware Checkerboard Sampling

Building on the concepts introduced in [2], [8], [15], [16], [34], we partition the pixels of I_{ref} into red–black grids using a checkerboard pattern. This arrangement allows simultaneous updates of hypotheses for black pixels using red pixels and vice versa. Unlike the method in [2], [8], [34], which samples from eight fixed positions, or the approach in [15], [16], which expands into structured region lines and adaptively samples better positions based on their previous multi-view matching costs, our method introduces an adaptively aware boundary-aware Octagram checkerboard propagation scheme that treats sampling as a search process guided by structure yet free to explore open regions across boundaries.

The structural region lines in [15], [16] consist of four V-shaped areas and four long strip areas, as shown in Figure 1. We build on these approaches by incorporating eight neighbouring structural region lines, referred to as Octagram neighbouring lines, while maintaining four distant structural region lines. This results in a total of twelve structural lines, shown in Figure 1, that propagate and mine for non-local optimal hypotheses. Crucially, the proposed boundary-aware search extends beyond the immediate, structure-constrained neighborhoods into more unconstrained open regions without abandoning the structural prior: the lines remain the organizing scaffold, but their sampling windows can shift across edges to the opposite side of boundaries. In practice, this adaptive extension increases the probability of encountering valid plane hypotheses when the central pixel is weakly textured, occluded, or lies on a discontinuity, while still preserving alignment with the underlying structural geometry. Non-local plane hypotheses therefore address texture-less, occluded, and non-Lambertian cases by leveraging neighbouring plane support instead of the central pixel, and the Octagram lines evaluate both local and non-local points to ensure accurate representation in the aggregated matching cost.

Existing methods such as ACMH [15] and ACMM [16] often misrepresent non-local plane hypotheses, leading to incomplete 3D models with holes and local inaccuracies. Our design improves depth estimation in such regions by adaptively extending the search along the same structural scaffolds into open areas where more informative support may exist, while retaining the structural lines as priors for coherence. The proposed sampling scheme thus uses structured region information to prioritize hypotheses with lower multi-view matching costs, yet permits boundary-crossing exploration so that viable planes are not missed when support lies just beyond the immediate neighborhood. By facilitating accurate propagation across shared regions and across boundaries, the Octagram lines yield more compact and reliable depth estimates with enhanced matching performance.

This adaptively aware boundary-preserving extension leads to strong regularization of 3D reconstructed points effectively reducing noise and suppressing outliers. As a result, the reconstructed results align more closely with expected geometric structures such as buildings with repetitive features or manufactured objects with sharp boundaries, as demonstrated in Figure 2, thereby enhancing overall model completeness, reliability, and accuracy.

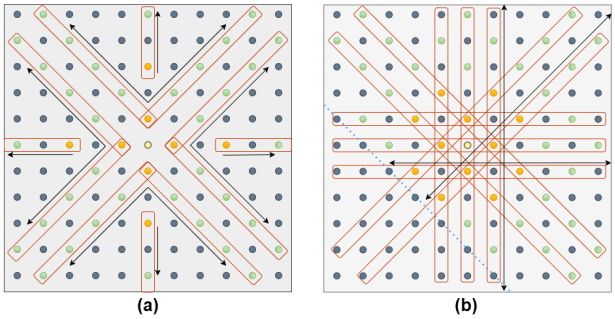


Fig. 2. Checkerboard propagation for sampling neighboring support plane hypotheses in textureless, occluded, or boundary regions. Dark gray dots indicate positions excluded from sampling, green dots denote valid structured-region candidates, orange dots represent starting neighbors and joint view probability points, and the yellow dot marks the center pixel under weak texture, occlusion, or boundary conditions. (a) ACMM employs Adaptive Checkerboard Propagation Matching [15], while (b) TPM-MVS applies boundary-aware Octagram propagation mining [36]. The boundary awareness extends the search across structured regions to the opposite side of boundaries, thereby capturing stronger edge responses.

D. Neighbouring Joint View Probability

To enhance the robustness of multi-view matching costs for each pixel, we leverage twelve structured hypotheses to infer the view weight of every neighbouring view. For a given pixel p, we compute its corresponding matching costs using the propagated hypotheses and represent them in a cost matrix $\mathbf{M_{cost}}$:

$$\mathbf{M_{cost}} = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,N-1} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ m_{12,1} & m_{12,2} & \cdots & m_{12,N-1} \end{bmatrix}, \quad (2)$$

where $m_{i,j}$ represents the matching cost for the i-th hypothesis h_i from the source image evaluated against the j-th view I_j from a set of N visible reference images. To refine the matching cost for the i-th hypothesis, the view selection prior for the j-th reference image is computed using neighbouring pixel positions (i.e., up, down, left, and right) through a joint-view probability framework as outlined in [15], [16].

In both ACMH and ACMM, four neighbouring pixel points in the cardinal directions are utilized to establish a view selection prior based on the outcomes of neighbouring plane hypotheses which are encoded as binary flags. The heuristic search then identifies candidate hypotheses that yield the minimum initial matching cost to be aggregated for the final matching cost, with each candidate hypothesis indicated by a binary flag. These four neighbouring points contribute to the view selection prior of the j-th reference image by indicating whether a particular view is likely to be correct for a given pixel p. This approach aggregates multiple hypotheses from adjacent points into a probability distribution function that guides view selection, ensuring that pixels with similar or consistent hypotheses reinforce one another while unreliable views are suppressed to promote spatial coherence in the reconstructed depth surface.

In the proposed method, structural region lines are defined such that they intersect around the given pixel p. Those intersecting points along these structural region lines serve as the basis for the view selection prior, ensuring that both non-local and local plane hypotheses are accurately selected for computing the aggregate matching cost. From these twelve points, we have predefined points which align with each distinctive structural line as illustrated in Figure 2. These predefined points are selected based on the binary flag outcome of the neighbouring hypothesis. Thus, each neighbouring hypothesis sample corresponds to a specific structural region line and intersects these neighbouring points, which act as the view selection priors. For example, the diagonal region line contains four intersecting points that are used as predefined points to support its plane hypotheses in the view selection process.

Furthermore, the selection probability of a candidate view depends on the visibility and consistency of its neighbouring views, ensuring a robust and coherent selection process. Each candidate view is influenced by a predefined set of neighbouring points. We employ an adaptive cost threshold and exponential weighting similar to those outlined in ACMH [15] and ACMM [16] to prioritize views with lower matching costs while mitigating unreliable hypotheses. The probability density function (PDF) is transformed into a cumulative distribution function (CDF) enabling efficient random sampling via binary search. This procedure resembles a Monte Carlo selection strategy by reinforcing depth estimation through probabilistic hypotheses evaluation. Ultimately, by integrating the neighbouring joint-view selection probability framework, our method dynamically selects views based on the visibility and consistency of adjacent neighbors, effectively reducing inconsistencies and avoiding isolated view selections.

In addition to these structured priors, we further incorporate a center re-anchoring mechanism that improves the stability of photometric consistency. For a given pixel p, the plane-induced homography H maps p to its corresponding source location $s_p = \Pi(H\tilde{p})$. Instead of directly relying on (p, s_p), we define a candidate set of neighbouring reference points $r \in \mathcal{N}(p)$, each with source mapping $s = \Pi(H\tilde{r})$. For each candidate, we compute a combined score:

$$S(r) = w_{\text{geo}}(r)\, w_{\text{ref}}(r)\, w_{\text{src}}(r)\, a_r(r)\, a_s(r), \quad (3)$$

where $w_{\text{geo}}(r) = \exp\!\left(-D^2(r)/(2\sigma_g^2)\right)$ encodes geometric proximity in both reference and source planes, $w_{\text{ref}}(r)$ and $w_{\text{src}}(r)$ measure texture similarity to the reference and source centers, and $a_r(r), a_s(r)$ are smooth border availability terms. The best anchor pair (p_r, p_s) is then chosen as:

$$(p_r, p_s) = \arg\max_{r \in \mathcal{N}(p)} \Big(S(r),\ \text{tie-break:}\ \min D^2(r) \Big). \quad (4)$$

This re-anchoring strategy is illustrated conceptually in Algorithm 1 and Figure 3, replaces the original window centers with (p_r, p_s) in the subsequent photometric matching, allowing the bilateral weights to be computed around more reliable anchor points. The idea is inspired by UAV aerial image stitching, where robust anchor points are selected to stabilize mosaicking [43], [44]. By adapting this concept to local multi-view stereo, we ensure that weak-texture, occluded, or boundary pixels are anchored to more stable local structures before evaluating photometric consistency.

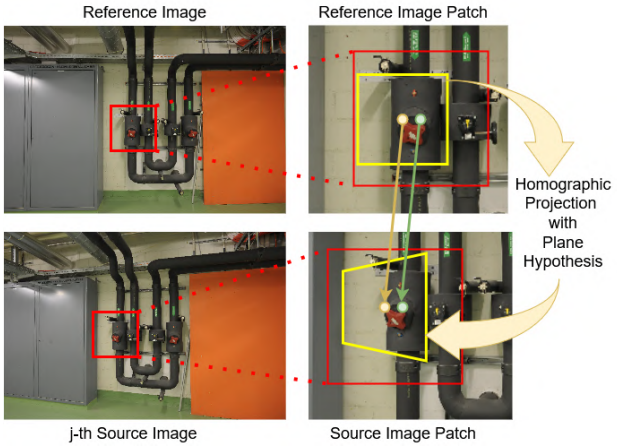


Fig. 3. Illustration of the re-anchoring strategy for dual bilateral weighting with photometric consistency. Beige dots indicate the center pixel points in the reference image and the j-th source image, while green dots represent the newly established re-anchoring points. These re-anchored points enhance the matching process by leveraging bilateral weights for improved consistency.

Algorithm 1 Re-anchoring strategy for dual bilateral weighted photometric matching (see Figure 3)

Require: Pixel p, homography H, reference image I_r, source image I_s, parameters σ_c, σ_g, R
Ensure: Best anchor pair (p_r, p_s)

```
Compute initial source projection s_p = Π(H p̃)
Initialize best score S* = −∞, best distance D²_min = ∞
for each candidate r ∈ N(p) do
    Map candidate to source: s = Π(H r̃)
    Compute geometric distance D²(r) = ‖r − p‖² + ‖s − s_p‖²
    Evaluate weights: w_geo, w_ref, w_src, a_r, a_s
    Compute score S(r) using Eq. (3)
    if S(r) > S* or (S(r) = S* and D²(r) < D²_min) then
        Update S* = S(r), D²_min = D²(r)
        Set p_r = r, p_s = s
    end if
end for
return (p_r, p_s)
```

Within each $m_{i,j}$ matching cost of the i-th hypothesis h_i from the source image evaluating it against the j-th view I_j, we then adopt a less restrictive weighting for the stereo view matching process. In [8], [15], [16], [19], [34], bilateral weighting is applied within the normalized cross-correlation function to compute matching costs. Here, the re-anchored centers are integrated into a dual bilateral weight adaptation of normalized cross-correlation to properly quantify photometric consistency between the source and reference view. This adaptation assists in handling occluded and textureless regions, ensuring that the cost remains reliable even when the original center pixel was unstable. The source and reference information can be denoted as $A = I_i'(p_i) = I_i(p_i) - \mu_i$ and $B = I_j'(p_j) = I_j(p_j) - \mu_j$ respectively. The center pixel and neighbouring pixel for the source I_i and reference view I_j are denoted as $(p_i, p_i + \Delta_i)$ and $(p_j, p_j + \Delta_j)$ respectively. The mean intensity of μ_i and μ_j are given by:

$$\mu_i = \frac{\sum_{p \in W} \omega_{ij}(p) \cdot I_i(p_i)}{\sum_{p \in W} \omega_{ij}(p)}, \tag{5}$$

$$\mu_j = \frac{\sum_{p \in W} \omega_{ij}(p) \cdot I_j(p_j)}{\sum_{p \in W} \omega_{ij}(p)}. \tag{6}$$

where W is the set of all pixel coordinates $(p.x, p.y)$ in the square window region for template matching. The dual bilateral weight adaptation of the normalized cross-correlation is then given as:

$$m_{i,j} = \frac{\sum_{p \in W} \omega_{ij}(p) \cdot A \cdot B}{\sqrt{\sum_{p \in W} \omega_{ij}(p) \cdot A^2} \cdot \sqrt{\sum_{p \in W} \omega_{ij}(p) \cdot B^2}}, \tag{7}$$

where $\omega_{ij}(p)$ is the unified bilateral weight composed of $\omega_r(p_{ij})$ and $\omega_s(p_{ij})$, defined as:

$$\omega_{ij}(p) = \omega_r(p_{ij}) \cdot \omega_s(p_{ij}), \tag{8}$$

$$\omega_r(p_{ij}) = \exp\left(-\frac{R(p_i)^2 + R(p_j)^2}{2\sigma_r^2}\right), \tag{9}$$

$$\omega_s(p_{ij}) = \exp\left(-\frac{S(p_i)^2 + S(p_j)^2}{2\sigma_s^2}\right). \tag{10}$$

Here, σ_s is the spatial distance and σ_r is the intensity range, set to $\{5.0, 3.0\}$ respectively. The dual terms $R(p_i), R(p_j)$ and $S(p_i), S(p_j)$ reinforce the weight when both source and reference exhibit consistent changes (e.g., aligned spatial offsets or similar intensity contrasts), thereby increasing confidence in shared structures across views. Conversely, when the differences oppose each other, the cross terms reduce the weight, suppressing inconsistent contributions. Together with the re-anchoring strategy, this mechanism selectively emphasizes correspondences supported by both images, improves robustness against noise and occlusions, and stabilizes photometric matching in textureless or boundary cases.

From the adaptation of the matching cost function to the view selection prior, we adopt a similar voting scheme to aggregate views from the cost matrix in order to determine their suitability. The core observation is that a poor view consistently exhibits high matching costs across its corresponding hypotheses. In contrast, a good view will have lower matching costs that tend to decrease as the algorithm iterates. The matching cost boundary for a good view $S_{\text{good}}(j)$ is defined as:

$$\tau(t) = \tau_0 \cdot e^{-\frac{t^2}{\alpha}}, \tag{11}$$

where t denotes the t-th iteration of propagation process, τ_0 is the initial matching cost threshold from the random initialization, and α is a constant. Furthermore, we introduce a fixed bad matching cost threshold τ_1 for matching costs which are greater than $\tau(t)$ ($\tau_1 > \tau(t)$). For a given view I_j, there must exist more than n_1 matching costs that satisfy $m_{i,j} < \tau(t)$ which indicates the set of good visible images $S_{\text{good}}(j)$.Additionally, there should be fewer than n_2 matching costs meeting this criteria $m_{i,j} > \tau_1$. The images that satisfy both these conditions are included in the view selection set $S(t)$ for iteration t. The inferred view selection set $S(t)$ may still include unstable views due to noise, varying viewpoints, or scale changes. To address this, the confidence of each matching cost $C(m_{i,j})$ is computed as:

$$C(m_{i,j}) = e^{-\frac{m_{i,j}^2}{2\beta^2}}, \tag{12}$$

where β is a constant that makes good views more distinctive. The view selection set $S(t)$ contains the reference images I_j's that meet the two conditions for a good view and $S_{\text{good}}(j)$ contains these good views. Using this confidence measure $C(m_{i,j})$, the normalized view weight of each selected view $w(I_j)$ can be defined as:

$$w(I_j) = \frac{1}{|S_{\text{good}}(j)|} \sum_{m_{i,j} \in S_{\text{good}}(j)} C(m_{i,j}), \quad I_j \in S(t). \quad (13)$$

To maintain consistency between t iterations, we incorporate the most influential view from the previous iteration denoted as v_{t-1} into the current view selection process. The updated view weight formula is:

$$w'(I_j) = \begin{cases} (1 + \mathbb{I}(I_j = v_{t-1})) \cdot w(I_j), & \text{if } I_j \in S(t); \\ 0.2 \cdot \mathbb{I}(I_j = v_{t-1}), & \text{else,} \end{cases} \quad (14)$$

where $\mathbb{I}(\cdot)$ is an indicator function that returns 1 ($\mathbb{I}(\text{true}) = 1$) when the condition is true and 0 ($\mathbb{I}(\text{false}) = 0$) otherwise. This adjustment improves the robustness of our view selection strategy. With the updated weights $w'(I_j)$, the aggregated photometric consistency cost for pixel p under hypothesis h_i is defined as:

$$m_{\text{photo}}(p, h_i) = \frac{\sum_{j=1}^{N-1} w'(I_j) \cdot m_{i,j}}{\sum_{j=1}^{N-1} w'(I_j)}. \quad (15)$$

Finally, the optimal estimate for pixel p is updated by selecting the hypothesis that yields the lowest aggregated multi-view cost.

E. Depth Refinement

Following each red-black iteration, the depth refinement step is introduced to enhance the diversity of the solution space. The depth and normal of a pixel p are analyzed under three scenarios: either one, neither, or both factors are close to the optimal solution [8], [15], [16], [34]. To increase the likelihood of sampling the correct solution, we generate two additional hypotheses by randomly initializing and perturbing the current estimate. We then combine these newly generated depths and normals with the current depth and normal to yield nine distinct hypotheses.

Each hypothesis in the above set will also be tested by (15). The hypotheses with the minimum aggregated cost is chosen as the final estimate for pixel p. The above propagation, multi-view matching cost evaluation, and refinement are repeated multiple times to get the final depth map for I_{ref}. In the end, a median filter of size 5×5 is applied to final depth maps as in [15], [16].

III. Multi-Scale Confidence-Guided Geometric Consistency

Incorporating a multi-scale approach, OCMH at the coarsest scale achieves more reliable depth estimates in texture-less regions, forming the basis of OCMM. However, photometric consistency faces difficulties when refining these depth estimates at finer scales. This section outlines the use of geometric consistency guidance, augmented with confidence measures, to effectively optimize these estimates. Furthermore, a detail restorer is utilized to address errors propagated from coarser scales.

A. Confidence-Guided Geometric Consistency

After obtaining reliable depth estimates for texture-less areas at the coarsest scale, which are propagated to finer scales using joint bilateral upsampling [19], [30], further refinement is essential. Our method employs upsampled depth maps from source images to impose geometric constraints on these estimates, reducing errors caused by photometric consistency. Inspired by [2], [15], [16], we adopt the forward-backward reprojection error to ensure geometric consistency. Considering the depth of pixel p in image I_i as $D_i(p)$ and the camera parameters $\mathbf{P}_i = [\mathbf{M}_i \mid \mathbf{p}_i]$ [29], the back-projected 3D point $X_i(p)$ is computed as:

$$X_i(p) = \mathbf{M}_i^{-1} \cdot \big(D_i(p) \cdot p - p_i\big). \quad (16)$$

The reprojection error $\Delta e_{i,j}$ between the reference image I_{ref} and the source image I_j for the i-th hypothesis is defined as:

$$\Delta e_{i,j} = \min\left\{\|P_{\text{ref}}\, X_j\big(P_j\, X_{\text{ref}}(p)\big) - p\|,\, \delta\right\}. \quad (17)$$

where δ is a truncation threshold (i.e., $\delta = 3$) used to handle occlusions and outliers, ensuring that reprojection errors exceeding this threshold are clamped to δ to maintain numerical stability and robustness. This formulation guarantees that the minimum reprojection error within the delta threshold range is considered as the minimum output preventing extreme errors from influencing the system (i.e., $0 \leq \Delta e_{i,j} \leq 3$).

Any pixel depth with a reprojection error exceeding this threshold is excluded as unreliable geometric information. This refinement is integrated into the geometric consistency cost, which is calculated as:

$$m_{\text{geo}}(p, h_i) = \frac{\sum_{j=1}^{N-1} w'(I_j) \cdot \big(m_{i,j} + \lambda \cdot \Delta e_{i,j}\big)}{\sum_{j=1}^{N-1} w'(I_j)}, \quad (18)$$

To strengthen the proposed confidence-guided multi-view matching cost, we introduce view confidence, defined using reprojection distance ξ_{geo}, depth relative error ξ_{d}, normal pinch error ξ_{n}, and matching cost $m_{photo}(p, h_i)$. The confidence components are computed as in [37]:

$$C_{\text{geo}} = e^{-\frac{\xi_{\text{geo}}^2}{2\,\sigma_{\text{geo}}^2}} = e^{-\frac{\|x_p - x_{p'}\|^2}{2\,\sigma_{\text{geo}}^2}}, \quad (19)$$

$$C_{\text{d}} = e^{-\frac{\xi_{\text{d}}^2}{2\,\sigma_{\text{d}}^2}} = e^{-\frac{\left((d_p - d_{p'})/d_p\right)^2}{2\,\sigma_{\text{d}}^2}}, \quad (20)$$

$$C_{\text{n}} = e^{-\frac{\xi_{\text{n}}^2}{2\,\sigma_{\text{n}}^2}} = e^{-\frac{\arccos^2\left(n_p \cdot n_{p'}\right)}{2\,\sigma_{\text{n}}^2}}, \quad (21)$$

$$C_{\text{c}} = e^{-\frac{m(p,h_i)^2}{2\,\sigma_{\text{c}}^2}}. \quad (22)$$

where σ_{geo}, σ_{d}, σ_{n}, and σ_{c} are constants for each confidence metric which set to (5.0, 0.05, 0.8, 0.5) respectively. The overall multiview confidence is:

$$C_{\text{g}}(p, h_i) = \frac{\Delta e_{i,j} \cdot \sum_{j=1}^{N-1} \left(C_{\text{geo}}^j \cdot C_{\text{d}}^j \cdot C_{\text{n}}^j \cdot C_{\text{c}}^j\right)}{N-1}. \quad (23)$$

This confidence is integrated into the matching cost as:

$$m_{\text{conf}}(p,h_i) = \frac{\sum_{j=1}^{N-1} w'(I_j)\cdot\big(m(i,j)+\lambda\cdot(1-C_{\text{g}}(p,h_i))\big)}{\sum_{j=1}^{N-1} w'(I_j)}, \tag{24}$$

where $C_{\text{g}}(p,h_i)$ represents the confidence for pixel p, $(1 - C_{\text{g}}(p,h_i))$ replaces the projection error $\Delta e_{i,j}$, and hypothesis h_i, ensuring robust depth estimation across scales. where $w'(I_j)$ represents the weight of the source image, $m_{i,j}$ denotes the initial matching cost, and λ balances photometric and geometric consistency, it is set to $\lambda = 0.2$. Recognizing that initial depth maps obtain may contain noise due to ambiguities and occlusions, geometric consistency addresses these limitations while refining depth maps iteratively across scales.

B. Depth Map Restorer

Multi-scale geometric consistency improves textureless regions but blurs fine details: coarse scales lose image cues, and a fixed OCMH patch (local-planar) struggles on thin structures [14], [31], [32]. Upsampling spreads reliable coarse estimates yet can corrupt boundaries [30]. We therefore add a detail restorer that targets thin/boundary areas by enforcing photometric consistency only where coarse-to-fine transfer is unreliable. After upsampling depth and normals to level l, we compute the initial photometric cost C^l_{init}, run the base MVS at this scale to obtain C^l_{photo}, and flag pixels with

$$C^l_{\text{init}}(p) - C^l_{\text{photo}}(p) > \xi, \tag{25}$$

replacing their hypotheses accordingly. This selective correction restores edges and thin structures while preserving textureless-region stability.

IV. Fusion

Following [2], [8], [15], [16], [34], we fuse all per-view depth maps into a single 3D point cloud. Each image is treated in turn as the reference: its depth map is back-projected to world coordinates, then reprojected into neighbouring views to seek correspondences. A match is declared *consistent* if it satisfies (i) relative depth difference $\epsilon \leq 0.01$, (ii) normal angle $\theta \leq 30^\circ$, and (iii) reprojection error $\psi \leq 2$ (as in [15], [16]). If at least $n \geq 2$ neighbouring views meet these criteria, the depth estimate is accepted; the corresponding 3D positions and normals from all consistent views are then averaged to yield the final fused point cloud.

V. Experiments

We evaluate our proposed method on a widely recognized dataset. The ETH3D benchmark [20], [35] includes training and test image datasets; however, we focus exclusively on the training dataset since we have access to the ground truth models to perform evaluation on the reconstructed model, using consistent parameter settings across all experiments for fair comparison [36]. All experiments were conducted on an HP Omen machine equipped with an Intel Core i7-10750H CPU and a GeForce RTX 2060 GPU. For the multi-hypotheses neighbouring joint-view selection scheme, the parameters were set as $\{\tau_0, \tau_1, \alpha, \beta, n_1, n_2\} = \{0.8, 1.2, 90, 0.3, 2, 3\}$. In the geometric consistency guidance strategy and detail restoration process, the parameters were $\{k, \eta, \delta, \lambda, \xi\} = \{3, 0.5, 3, 0.2, 0.1\}$, with $\sigma_{\text{geo}}, \sigma_{\text{d}}, \sigma_{\text{n}}, \sigma_{\text{c}} = \{5.0, 0.05, 0.8, 0.5\}$. Additionally, to speed up the computation of the matching cost, only every other row and column in the window was used, following the approach in [1], [2], [8], [15], [16], [34].

A. Datasets and Metrics

We evaluate point-cloud quality following ETH3D [20], reporting accuracy (alignment to ground truth), completeness (coverage), and their harmonic mean (F1). Evaluations use ground-truth scans for the training split or the online server for the test split. Metrics are computed at thresholds $\rho_e \in \{1, 2, 5, 10, 20, 50\}$ cm: smaller (e.g., 2 cm) probe fine-detail fidelity (thin/small structures), while larger (e.g., 10 cm) emphasize overall geometric integrity [20]. The benchmark specifically stresses textureless regions, occlusions, thin/repetitive structures, and non-Lambertian surfaces [20].

ETH3D [20], [35] provides high-resolution multi-view imagery (6048×4032) for diverse indoor/outdoor scenes. The training set includes ground-truth depth; the test set omits ground truth and is scored via the online system. This dataset remains a strong proxy for real-world MVS difficulty, enabling concise, apples-to-apples comparisons across methods under challenging conditions.

B. Point Cloud Evaluation Results

The performance of the proposed TPM-MVS method was evaluated against existing approaches using public benchmark ETH3D dataset to assess point cloud reconstruction quality. The results are detailed in Table I includes evaluations under 2 cm threshold. Figure 4 provides visual comparisons and results of the reconstructed point clouds under ETH3D benchmark. The analysis focused on high-resolution training datasets from the ETH3D comparing the proposed method with COLMAP, ACMH, ACMM, ACMMP, MAR-MVS, UGNet, EPNet, and OPM-MVS. Metrics such as accuracy, completeness, and F1 scores were used to evaluate performance in ETH3D benchmark. At a 2 cm threshold in the ETH3D benchmark informs us of the reconstruction accuracy of MVS performance under stringent conditions.

Under a threshold of 2 cm in the ETH3D training dataset under courtyard performance, TPM-MVS achieved completeness and an F1-score of 94.34% and 90.07% in Table I, outperforming COLMAP's 73.47% and 80.49%, ACMM's 82.85% and 86.89%, ACMMP's 85.87% and 88.47%, MAR-MVS's 89.87% and 87.46%, and UGNet's 69.88% and 71.65%, which indicates improved reconstruction in texture-less regions. Similarly, in the outdoor dataset of meadow in Table I, TPM-MVS obtained completeness and an F1-score 75.49% and 77.10%, surpassing COLMAP's 34.39% and 49.96%, ACMM's 60.74% and 71.49%, ACMMP's 62.61% and 72.77%, MAR-MVS's 58.33% and 68.25%, and UGNet's 51.87% and 56.72%,

reinforcing its capability in maintaining dense point cloud generation. When it comes to accuracy wise, our prior MVS baseline [36] performs much better than the proposed method. Where TPM-MVS shines is in the completeness of the reconstructed models with courtyard's 94.34%, delivery area's 93.91%, electro's 89.34%, meadow's 75.49%, office's 82.52%, pipes's 83.34%, playground's 83.53%, and terrace's 89.62%. This textureless-resilient behavior yields robust large-scale reconstructions. Overall, TPM-MVS attains excellent completeness and F1-scores on 5 of 8 scenes while maintaining accuracy comparable to strong MVS baselines. For outdoor scenes, it achieves comparable completeness to competing methods yet produces more detailed and complete point clouds in challenging regions (Fig. 4).

VI. Conclusion

The proposed method augments our prior MVS baseline with two key innovations: (i) an anchor-guided dual bilateral weighting for the photometric matching that re-anchors the matching window to reduce reference-view inconsistencies, and (ii) a boundary-aware structurally guided sampling strategy that adaptively extends checkerboard propagation across edges to harvest non-local plane support. Embedded in a multi-scale coarse-to-fine framework, these components optimize hypothesis propagation while enforcing robust photometric and geometric consistency yielding accurate, more complete depth maps in texture-less, occluded, and non-Lambertian regions. On the ETH3D training dataset, the approach consistently improves completeness and preserves fine detail outperforming competing methods in scenes with complex geometry and large texture-less areas. While ACMMP (i.e., ACMM with planar priors assist) remains top on the overall set over our prior OPM-MVS, the present TPM-MVS delivers state-of-the-art reconstruction quality among non-learned pipelines surpassing ACMH, ACMM, ACMMP, and the surveyed MVSNet variants.

Acknowledgment

The authors received no specific funding for this work. Ray L. Khuboni gratefully acknowledges Prof. Hongjun Xu for his invaluable guidance, feedback, and support throughout this research.

References

[1] Y. Furukawa and J. Ponce, "Accurate dense and robust multiview stereopsis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362-1376, Aug. 2010.

[2] S. Galliani, K. Lasinger and K. Schindler, "Massively parallel multiview stereopsis by surface normal diffusion," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 873-881, 2015.

[3] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1-8, 2007.

[4] Q. Shan, R. Adams, B. Curless, Y. Furukawa, and S. M. Seitz, "The visual turing test for scene reconstruction," in *Proc. Int. Conf. 3D Vis.*, pp. 25-32, 2013.

[5] Q. Shan, B. Curless, Y. Furukawa, C. Hernandez, and S. M. Seitz, "Occluding contours for multi-view stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 4002-4009, 2014.

[6] S. Shen, "Accurate multiple view 3D reconstruction using patch-based stereo for large-scale scenes," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1901-1914, May 2013.

[7] E. Zheng, E. Dunn, V. Jojic, and J.-M. Frahm, "PatchMatch based joint view selection and depthmap estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1510-1517, 2014.

[8] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis.*, pp. 501-518, 2016.

[9] J. Wei, B. Resch, and H. Lensch, "Multi-view depth map estimation with cross-view consistency," in *Proc. Brit. Mach. Vis. Conf.*, pp. 1-13, 2014.

[10] M. Pollefeys *et al.*, "Detailed real-time urban 3D reconstruction from video," *Int. J. Comput. Vis.*, vol. 78, no. 2, pp. 143-167, Jul. 2008.

[11] P. Merrell *et al.*, "Real-time visibility-based fusion of depth maps," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1-8, 2007.

[12] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," in *Proc. ACM SIGGRAPH Conf.*, pp. 24:1-24:11, 2009.

[13] C. R. M. Bleyer and C. Rother, "PatchMatch stereo - Stereo matching with slanted support windows," in *Proc. Brit. Mach. Vis. Conf.*, pp. 14.1-14.11, 2011.

[14] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 9, pp. 920-932, Sep. 1994.

[15] Q. Xu and S. Yao, "ACMH: Adaptive Checkerboard Propagation for Multi-View Stereo," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 8738–8747. doi: 10.1109/CVPR.2019.00894.

[16] Q. Xu and W. Tao, "Multi-scale geometric consistency guided multi-view stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 5478-5487, 2019.

[17] Q. Xu and W. Tao, "Planar prior assisted PatchMatch multi-view stereo," in *Proc. AAAI Conf. Artif. Intell.*, pp. 12516-12523, 2020.

[18] Q. Xu, W. Kong, W. Tao, and M. Pollefeys, "Multi-Scale Geometric Consistency Guided and Planar Prior Assisted Multi-View Stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4945–4963, 1 April 2023, doi: 10.1109/TPAMI.2022.3200074.

[19] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, "Bilateral Filtering: Theory and Applications," *Foundations and Trends in Computer Graphics and Vision*, vol. 4, no. 1, pp. 1–73, 2009. doi: 10.1561/0600000020.

[20] T. Schöps *et al.*, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2538-2547, 2017.

[21] O. Faugeras and R. Keriven, "Variational principles surface evolution PDEs level set methods and the stereo problem," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 336-344, Mar. 1998.

[22] G. Vogiatzis, C. H. Esteban, P. H. S. Torr, and R. Cipolla, "Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2241-2246, Dec. 2007.

[23] S. N. Sinha, P. Mordohai, and M. Pollefeys, "Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh", *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1-8, 2007.

[24] C. H. Esteban and F. Schmitt, "Silhouette and stereo fusion for 3D object modeling", *Comput. Vis. Image Understanding*, vol. 96, no. 3, pp. 367-392, 2004.

[25] V. H. Hiep, R. Keriven, P. Labatut, and J. Pons, "Towards high-resolution large-scale multi-view stereo", *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1430-1437, 2009.

[26] D. Cremers and K. Kolev, "Multiview stereo and silhouette consistency via convex functionals over convex domains", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1161-1174, Jun. 2011.

[27] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 418-433, Mar. 2005.

[28] A. Romanoni and M. Matteucci, "TAPA-MVS: Textureless-aware PatchMatch multi-view stereo", *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 10 412-10 421, 2019.

[29] R. Hartley and A. Zisserman, *Multiple View Geometry in Comput. Vision*, Cambridge, U.K.: Cambridge Univ. Press, 2004.

[30] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling", *ACM Trans. Graph.*, vol. 26, no. 3, 2007.

TABLE I

POINT CLOUD EVALUATION ON SELECTED ETH3D HIGH-RESOLUTION MULTI-VIEW *training* SCENES AT 2 CM. ENTRIES ARE ACCURACY / COMPLETENESS / F1 (%). RESULTS FROM [20], [35]. BOLD = BEST F1, RED = SECOND-BEST F1.

Method	Metric	Courtyard	Delivery Area	Electro	Meadow	Office	Pipes	Playground	Terrace
COLMAP [8], [34]	Acc	88.98	**92.86**	95.51	**91.27**	**94.11**	97.77	**84.22**	**96.79**
	Comp	73.47	67.21	62.13	34.39	31.60	34.24	44.90	75.67
	F1	80.49	77.98	75.29	49.96	47.32	50.72	58.57	84.94
ACMH [15]	Acc	86.45	87.16	91.47	89.74	93.55	**98.50**	77.96	93.49
	Comp	78.92	78.26	67.65	30.98	38.25	34.59	61.37	79.74
	F1	82.51	82.47	77.78	46.06	54.30	51.20	68.68	86.07
ACMM [16]	Acc	**91.35**	89.93	96.34	86.88	87.32	96.63	83.32	96.19
	Comp	82.85	77.76	77.70	60.74	49.29	53.97	65.86	84.13
	F1	86.89	83.40	86.02	71.49	63.01	69.26	73.57	89.76
ACMMP [18]	Acc	91.24	91.60	**96.49**	86.87	87.24	96.46	82.87	96.35
	Comp	85.87	85.58	81.42	62.61	78.53	65.79	68.80	85.75
	F1	88.47	**88.49**	88.31	72.77	**82.65**	78.22	**75.18**	**90.74**
MAR-MVS [42]	Acc	85.18	85.85	89.13	82.25	71.28	88.89	69.69	89.52
	Comp	89.87	90.58	79.33	58.33	59.24	67.98	74.83	86.95
	F1	87.46	88.15	83.95	68.25	64.70	77.04	72.17	88.22
UGNet [40]	Acc	73.52	83.91	86.64	62.57	85.82	89.63	53.18	88.28
	Comp	69.88	82.58	66.43	51.87	61.74	62.01	45.46	84.24
	F1	71.65	83.24	75.20	56.72	71.82	73.30	49.02	86.22
EPNet [38]	Acc	89.51	77.20	88.35	74.29	76.31	79.22	59.63	86.49
	Comp	88.31	90.14	79.43	61.26	77.26	70.08	70.06	87.82
	F1	88.90	83.17	83.65	67.15	76.78	74.37	64.43	87.15
OPM-MVS [36]	Acc	90.86	89.35	95.07	86.87	91.07	97.29	79.13	94.12
	Comp	85.30	84.34	82.82	55.39	54.09	60.14	70.04	85.74
	F1	87.99	86.77	**88.52**	67.65	67.87	74.33	74.31	89.73
TPM-MVS	Acc	86.18	83.43	87.57	78.77	69.75	84.63	62.24	89.88
	Comp	**94.34**	**93.91**	**89.34**	**75.49**	**82.52**	**83.34**	**83.53**	**89.62**
	F1	**90.07**	88.36	88.45	**77.10**	75.60	**83.98**	71.33	89.75

[31] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 1073-1079, Jul. 2009.

[32] K.-J. Yoon and I.-S. Kweon, "Locally adaptive support-weight approach for visual correspondence search", *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 924-931, 2005.

[33] J. Yan et al., "Dense hybrid recurrent multi-view stereo net with dynamic consistency checking", *Proc. Eur. Conf. Comput. Vis.*, pp. 674-689, 2020.

[34] J. L. Schönberger and J. Frahm, "Structure-from-motion revisited", *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 4104-4113, 2016.

[35] T. Schöps et al., "ETH3D benchmark", [online] Available: https://www.eth3d.net. [Accessed: 22 April 2025].

[36] R. L. Khuboni and H. Xu, "Octagram Propagation Matching for Multi-Scale View Stereopsis (OPM-MVS)," *IEEE Access*, vol. 13, pp. 86203–86217, 2025. doi: 10.1109/ACCESS.2025.3569913.

[37] C. Fu, N. Huang, Z. Huang, Y. Liao, X. Xiong, X. Zhang, and S. Cai, "Confidence-Guided Planar-Recovering Multiview Stereo for Weakly Textured Plane of High-Resolution Image Scenes," *Remote Sensing*, vol. 15, no. 9, p. 2474, 2023.

[38] W. Su and W. Tao, "Efficient Edge-Preserving Multi-View Stereo Network for Depth Estimation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, pp. 2348–2356, 2023.

[39] L. Zhou, Z. Zhang, H. Jiang, H. Sun, H. Bao, and G. Zhang, "DP-MVS: Detail Preserving Multi-View Surface Reconstruction of Large-Scale Scenes," *Remote Sensing*, vol. 13, no. 22, article 4569, 2021, doi:10.3390/rs13224569.

[40] W. Su, Q. Xu, and W. Tao, "Uncertainty Guided Multi-View Stereo Network for Depth Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7796–7808, Nov. 2022, doi:10.1109/TCSVT.2022.3183836.

[41] Q. Yan, J. Kang, T. Xiao, H. Liu, and F. Deng, "MVP-Stereo: A Parallel Multi-View Patchmatch Stereo Method with Dilation Matching for Photogrammetric Application," *Remote Sensing*, vol. 16, no. 6, article 964, 2024, doi:10.3390/rs16060964.

[42] Z. Xu, Y. Liu, X. Shi, Y. Wang, and Y. Zheng, "MARMVS: Matching Ambiguity Reduced Multiple View Stereo for Efficient Large Scale Scene Reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[43] R. Shao, C. Du, H. Chen, and J. Li, "Fast Anchor Point Matching for Emergency UAV Image Stitching Using Position and Pose Information," *Sensors*, vol. 20, no. 7, Art. no. 2007, Apr. 2020. doi: 10.3390/s20072007.

[44] M. Brown and D. G. Lowe, "Automatic Panoramic Image Stitching using Invariant Features," *Int. J. Comput. Vis.*, vol. 74, no. 1, pp. 59-73, Aug. 2007. doi: 10.1007/s11263-006-0002-3.

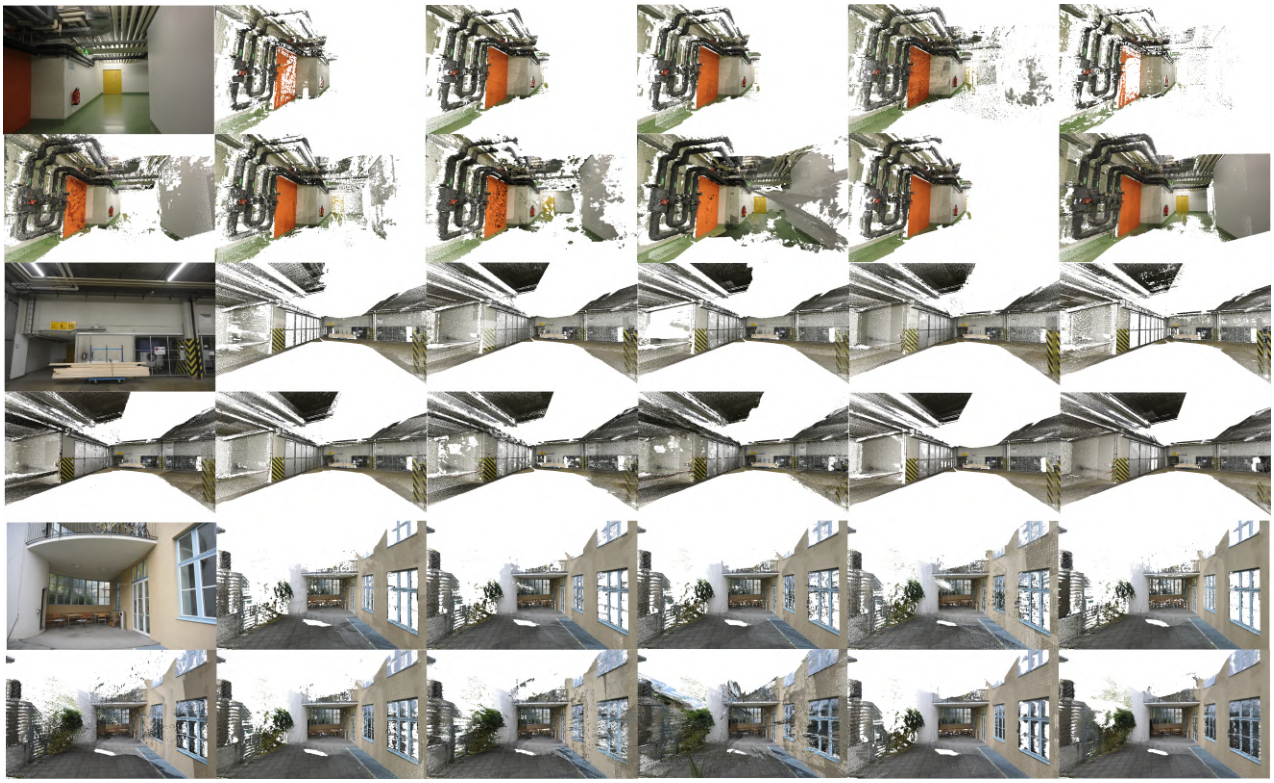


Fig. 4. The point clouds of the High-Resolution Multi-View Training dataset of ETH3D benchmark. The results are published on the ETH3D website [20], [35]. From top to bottom: pipes, delivery area, and terrace reconstructions. From left to right: Image view, (first-row) ACMH [15], ACMP [17], ACMM [16], ACMMP [18], COLMAP [8], [34], (second-row) MAR-MVS [42], TAPA-MVS [28], UGNet [40], EPNet [38], OPM-MVS [36] and TPM-MVS (proposed method).

Investigating Adaptive Fine-Tuning Strategies for Isolated Sign Language Recognition

1st Marc Marais
Computer Science
Rhodes University
Grahamstown, South Africa
marc.marais@ru.ac.za

2nd Dane Brown
Computer Science
Rhodes University
Grahamstown, South Africa
d.brown@ru.ac.za

3rd James Connan
Computer Science
Rhodes University
Grahamstown, South Africa
j.connan@ru.ac.za

***Abstract*—This paper explores adaptive fine-tuning strategies for isolated Sign Language Recognition (SLR) using the X-CLIP architecture, a vision–language model extended for video understanding. Isolated SLR presents significant challenges due to signer variability, spatiotemporal complexity, and limited training data. To address these challenges, we propose a sequential adaptive fine-tuning strategy, where layers of the pretrained X-CLIP backbone are progressively unfrozen when validation loss plateaus. Experiments were conducted on the large-scale isolated signer-independent AUTSL dataset to evaluate layer step-size sensitivity and compare fine-tuning approaches. Results demonstrate that adaptive fine-tuning utilising a step size of one consistently outperformed single-phase fine-tuning and full transfer learning, achieving a top test accuracy of 91.10%. The findings highlight the benefits of gradual adaptation for leveraging pretrained vision–language models in isolated signer-independent SLR and provide insights for extending this approach to low-resource sign languages and extending the framework towards continuous SLR. Beyond technical improvements, this work advances the broader goal of sustainable AI by maximising knowledge transfer, minimising training cost, and enabling accessibility for Deaf and Hard of Hearing communities.**

***Index Terms*—isolated sign language recognition, adaptive fine-tuning, transfer learning, vision-language models**

I. Introduction

Sign Language is the primary means of communication for the Deaf and Hard of Hearing community through a range of dynamic hand and body gestures. The structure of sign language contains its own grammatical rules, syntax, and vocabulary. Furthermore, the complexity of sign language is enhanced as it is not a universal language, containing significant variations between regions and cultures [1].

From a technical aspect, sign language is represented using visual spatiotemporal gestures that combine both manual and non-manual information for communication. Manual information focuses on the shape, orientation, position, and motion of the hands. Whereas, non-manual information consists of body posture, arm movements, eye gaze, lip shape, and facial expressions [2]. These gestures can represent letters, simple words, expressions and phrases with complex grammatical structures.

This work was undertaken in the Distributed Multimedia CoE at Rhodes University.

Sign Language Recognition (SLR) aims to utilise computer vision to develop machine learning and deep learning algorithms to translate sign language video to text, also referred to as sign glosses[1]. Additionally, SLR is split into two categories: isolated and continuous SLR. Isolated SLR recognise sign movements where a single video corresponds to only one gloss. In contrast, continuous SLR detects a sequence of signs from a single video corresponding to a sequence of glosses [4].

Naturally, sign language signers exhibit subtle gestural variations, which result in variations between signers, commonly referred to as the signer-independence problem [5]. The variability among signers encompasses multiple factors, namely:

- differences in hand configuration, movement patterns, and spatial positioning
- varying signing speeds due to intra/inter-personal variability and,
- diverse learning backgrounds and proficiency levels

These variations highlight the need for a robust SLR system that successfully handles the variations between signers.

Isolated SLR is part of the broader video recognition, which focuses on modelling both spatial and temporal information. Modelling both spatial and temporal information enables the model to learn spatial features about each sign as well as the dynamic temporal movements across the frames of the video.

Within the visual-textual joint representation space, contrastive language-image pretaining has shown promising success [6]. Recently, pretrained language-image models have been adapted to effectively approach video recognition tasks, utilising a cross-frame attention mechanism to capture long-range dependencies between frames, X-CLIP [7]. The X-CLIP module is lightweight, allowing for seamless transfer learning and fine-tuning for specific video recognition tasks.

For isolated SLR, X-CLIP's ability to model visual-textual correspondences is pivotal towards translating sign gestures to glosses. The cross-frame attention mechanism effectively captures temporal dependencies in sign movements, while the pretrained visual-textual representations provide robust feature

[1]A label representing a word or phrase associated with a sign language video [3]

extraction capabilities. By combining transfer learning initialisation with fine-tuning strategies, X-CLIP can be effectively trained for SLR.

This paper notably contributes towards:

- Advanced adaptive fine-tuning combined with transfer learning.
- The application of X-CLIP pre-trained vision transformer for isolated SLR.
- Enabling low-resource isolated SLR without reliance on multimodal data sources.

Subsequently, this paper focuses on isolated SLR on the AUTSL dataset, utilising the X-CLIP model combined with adaptive fine-tuning strategies and transfer learning. In Africa, particularly South Africa, where multiple dialects of sign language are present, which largely remain under-documented and have limited data to train models, methods such as fine-tuning are crucial for developing inclusive technologies.

The rest of the paper is structured as follows: Section II provides an overview of related work in the field of SLR. Followed by Sections III and IV, which detail the proposed fine-tuning technique and model architecture, along with the dataset and training implementation details. Finally, the results are provide and discussed in Section V, along with a conclusion in Section VI

II. Related Works

Traditional machine learning methods typically combine feature extraction techniques such as Principal Component Analysis, Linear Discriminant Analysis, or Histogram of Oriented Gradients, combined with a classification algorithm, namely support vector machines [8]. However, traditional machine learning approaches are limited due to neglecting temporal information, which is vital for isolated SLR and action recognition systems that require spatial and temporal components [2].

Subsequently, there has been a shift towards deep learning algorithms that include spatial and temporal information. Commonly implemented deep learning algorithms include convolutional neural networks (CNNs), recurrent neural networks (RNNs) and more recently, vision transformers [9].



(a) Pose estimation. (b) Segmented hands. (c) Raw frame.

Fig. 1: Different feature extraction techniques [10]. Fig. 1a highlights the hand keypoints extracted, Fig. 1b segments both the left and right hand against a black background, and Fig. 1c shows the original frame.

Various feature extraction techniques for SLR have been explored, including pose estimation, hand segmentation and raw full image input on the LSA64 dataset, as seen in Figure 1 [10]. The authors found that raw image data using a Pruned VGG architecture achieved the best performance, closely followed by the pose estimation combined with a 1D CNN. Segmenting the hands performed poorly due to the lack of spatial awareness of the rest of the signer's body. On the Ankara University Turkish Sign Language (AUTSL) signer-independent isolated SLR dataset [11], pose flow and self-attention have been proposed using square hand crops for manual feature extraction [12]. The following models are considered in their study: a Video Transformer Network (VTN), a VTN with hand crops and a VTN with hand crops and pose flow. ResNet-34 [13] pre-trained with ImageNet weights [14] was used as the backbone for the VTN 2D feature extractor. The best-performing model was the VTN combined with hand crops and pose flow, achieving an accuracy of 92.92% on the unseen test set. Notably, a significant accuracy improvement over the vanilla VTN by 8.10%.

Isolated SLR was evaluated on AUTSL and WLASL3000 datasets using vision-based approaches such as ResNet50-I3D and TimeSformer, and the pose-based SPOTER architecture (sign pose-based transformer) approaches [15]. Vision models were pretrained on Kinetics-400, while SPOTER utilised OpenPose and MMPose for skeletal extraction. On AUTSL, ResNet50-I3D achieved 92.86% accuracy, TimeSformer 85.89%, and SPOTER 84.90% with MMPose and 78.89% with OpenPose. Their neural ensemble approach achieved top performance with 96.37% on AUTSL and 70.12% on WLASL3000. Transfer learning from AUTSL to WLASL3000 improved TimeSformer accuracy by 15.76% (36.79% to 52.55%), though the reverse transfer showed no improvement due to AUTSL's greater signer and background variation.

Multi-phase fine-tuning has been explored on the continuous SLR RWTH-PHOENIX-Weather Multisigner 2014 dataset [16]. The authors compared single-phase vs multi-phase fine-tuning strategies using GoogleNet with Inception V3 modules. With single-phase fine-tuning, the k topmost layers are fine-tuned while the remaining layers' weights are kept frozen at their source network values. In contrast, the multi-phase approach sequentially trains the $top-k$ layers with a step-size s, which is the number of layers unfrozen at each sequential phase. The model was trained as an image classification problem, and the authors found that multi-phase fine-tuning results in higher classification accuracy, achieving a top-1 accuracy of 41.00% compared to the 30.00% when using single-phase fine-tuning. However, further exploration is needed to test the fine-tuning strategies on isolated SLR with temporal information.

III. Methods & Materials

In this section, we provide an overview of transfer learning and our adaptive fine-tuning strategy, in Sections III-A and III-B. Additionally, the X-CLIP architecture is explained in Section III-C.

A. Transfer Learning

Transfer learning is the methodology that leverages knowledge learnt in a source domain to improve a model in a target domain [17]. Traditional machine learning is strict in that training and testing data must originate from the same distribution. However, transfer learning violates this assumption due to the high cost and difficulty in collecting large labelled datasets for a new domain. Subsequently, leveraging transfer learning is valuable in scenarios where a model is needed for a new related domain, but training data is limited or costly to obtain. This is particularly useful in the SLR domain, as training data is limited [18]. Therefore, utilising transfer learning with models trained on a similar domain, such as action recognition, would significantly benefit model performance.

B. Adaptive Fine-Tuning

Fine-tuning is a technique of adapting a pre-trained model for solving a specific task and then fine-tuning certain layers of the model network for a related but different target task [19]. The model backbone layer weights are initialised with weights w_0 from a source network trained on a different dataset using transfer learning. The last classification layer is not initialised with weights. The layer weights of the network are subsequently fine-tuned using a training dataset from the target domain, in this case, isolated SLR. Typically, in single-phase fine-tuning, the k topmost layers are fine-tuned while keeping the rest of the model layers' weights frozen with their source network values [16].

Our adaptive fine-tuning phase trains the top-k layers in the model by implementing a sequential training strategy with a step-size s, resulting in (k/s) phases[2]. The initial phase fine-tunes the top-s layers, and in each subsequent phase, s more layers are fine-tuned. Within each phase, training proceeds until validation loss plateaus for P epochs. When this occurs, the next s layers are unfrozen; if no layers remain, fine-tuning ends. The sequential adaptive fine-tuning method is given in Algorithm 1.

C. X-CLIP Architecture

The X-CLIP model is an extension of CLIP (Contrastive Language–Image Pretraining) into the video domain. CLIP was introduced as a dual-encoder framework trained on large-scale image–text pairs with a contrastive objective [20]. Images are mapped into a latent embedding space using a visual encoder, while the text encoder projects natural language descriptions into the same space. Contrastive loss is used to ensure that image and text embeddings of matching pairs are aligned while mismatched pairs are pushed apart.

Subsequently, X-CLIP builds on this foundation by adapting CLIP's image–text alignment capabilities for video recognition [7]. Rather than relying on one-hot labels, which provide only categorical supervision, X-CLIP leverages text as supervision to incorporate richer semantic information. As seen in Figure 2, the model jointly trains a video encoder and a text encoder

[2]Where s is a factor of k.

Algorithm 1 Adaptive Fine-Tuning Strategy.

Require: Training set D_{train}, validation set D_{val}, model with backbone + classification head

Unfreeze the first block of backbone layers and the classification head. Begin Training.
$best_val \leftarrow +\infty$, $patience \leftarrow 0$
for each epoch **do**
  Train on D_{train} (only unfrozen layers)
  $v \leftarrow$ validation loss on D_{val}
  if $v < best_val$ **then**
    $best_val \leftarrow v$, $patience \leftarrow 0$
  else
    $patience \leftarrow patience + 1$
  end if
  if $patience \geq P$ and more layers remain frozen **then**
    Unfreeze next block of backbone layers
    $patience \leftarrow 0$
  end if
end for
return fine-tuned model

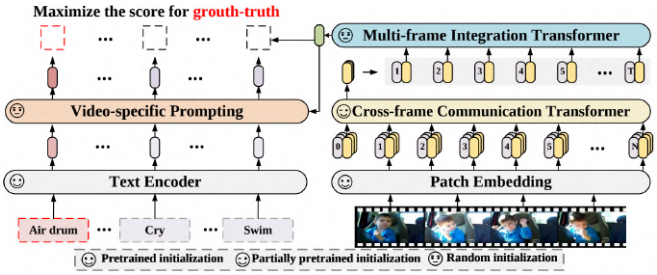


Fig. 2: Overview of X-CLIP architecture [7].

to output aligned embeddings, ensuring that the video representation is semantically close to its corresponding textual label.

The video encoder introduces two cascaded transformer modules, namely the Cross-Frame Communication Transformer (CCT) and Multi-Frame Integration Transformer (MIT). The CCT module introduces a cross-frame attention mechanism to process frame-level features extracted by a pretrained language-image model. The novelty of the X-CLIP model lies in the cross-frame attention, which combines cross-frame fusion attention and intra-frame diffusion attention along with a feed-forward network. The MIT module then aggregates frame representations into a single video-level embedding using temporal self-attention and average pooling. In total, the X-CLIP backbone has 12 video and text encoder blocks and therefore 12 layers to utilise within the fine-tuning process.

CLIP's pretrained test encoder is expanded with a video-specific prompting scheme for the X-CLIP text encoder. Instead of relying on manually designed prompts, the model learns task-adaptive prompts that condition the textual representation on the visual content. This dynamic prompting makes the textual supervision more discriminative in the video

setting.

IV. Experimental Setup

This section details the AUTSL dataset along with the metrics used to gauge model performance for isolated SLR in Sections IV-A and IV-B. Additionally, the training implementation pipeline is given in Section IV-C.

A. Dataset Description

The AUTSL dataset is a large-scale Turkish resource for signer-independent SLR, comprising 38,336 videos of 226 isolated signs performed by 43 different signers [11]. The recordings span diverse conditions, including varying backgrounds, environments, postures, and spatial positions, with data collected both indoors and outdoors, as demonstrated in Figure 3. Notably, in single-handed signs, the non-dominant hand is still present, either resting on the lap or positioned at the side, depending on the signer's posture.



Fig. 3: Sample video snapshots taken from the AUTSL dataset, showcasing the significant signer and background variation [11].

A sample imbalance exists across both the number of signs per class and the number of signs performed per signer. The number of samples for each signer ranges from 435 samples to 2697 samples, with an average number of samples per signer of 843. The number of samples for each sign ranges between 90 and 127, with an average of 124 samples. The signs *Sugar* and *Okay* are the two most underrepresented classes, with 90 and 95 samples each.

B. Evaluation Metrics

Model performance in SLR is typically assessed using a range of metrics, with accuracy being the most widely applied in isolated SLR. In this study, we report both accuracy and the F-score. Accuracy, given in Equation 1, quantifies the proportion of correctly predicted samples relative to the total classification attempts [21].

$$Accuracy = \frac{\text{number of correct classifications}}{\text{total number of classifications attempted}} \quad (1)$$

To provide a more balanced evaluation, particularly in the presence of class imbalance, we also employ the F-score, which integrates precision and recall into their harmonic mean, detailed in Equation 2 [21].

$$F\text{-}score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2)$$

C. Training Implementation

In this work, we adopt a pretrained X-CLIP model[3], which uses a ViT-B/16 backbone and is trained on the large-scale Kinetics-600 dataset, an extension of the original Kinetics 400 dataset for human action recognition [22], [23]. Each video is represented by randomly sampling 16 frames using uniform segments with a resolution of 224×224. The original classification head is replaced with a custom linear head composed of a projection to 512 dimensions, batch normalisation, ReLU activation, dropout, and a final linear layer mapping to the number of gloss classes.

A learning rate of 5×10^{-4} is applied to the X-CLIP model using a batch size of 8, based on manual inspection and computational processing time as informed by the literature. To stabilise optimisation, the model backbone is initially frozen, and an adaptive unfreezing strategy progressively releases transformer layers once validation loss plateaus, as explained in Section III-B. The maximum epochs is set to 500 to allow the model to unlock a significant number of its layers during training, effectively measuring the effect of adaptive unfreezing.

`FocalLoss` is implemented as the loss function, extending cross-entropy loss by focusing the optimisation on harder, often misclassified samples and down-weighting easy examples [24]. Thus making it particularly effective for the class sample imbalance as seen in the AUTSL dataset. The formulation of focal loss is given in Equation 3.

$$\mathcal{L}_{\text{focal}}(p_t) = -\alpha(1 - p_t)^{\gamma} \log(p_t), \quad (3)$$

where p_t is the predicted probability for the true class, α is a balancing factor, and γ controls the focusing strength. For optimisation, the `AdamW` algorithm with a weight decay coefficient of 1×10^{-5} is utilised. Compared to the `Adam` optimiser, `AdamW` decouples weight decay from the gradient update, leading to more stable convergence and improved model generalisation [25]. Furthermore, layer-wise learning rate decay is applied, assigning higher learning rates to later transformer blocks and the classifier head, and lower rates to earlier layers.

A ReduceLROnPlateau scheduler reduces the learning rate when validation loss saturates. To improve model generalisation ability, data augmentation is applied to the training dataset split, which includes random probabilistic resized cropping, horizontal flipping, affine transformations, colour jittering, and grayscale. *Early Stopping* with a patience value of 40 is employed, to ensure that all model layers are unfrozen to maximise model learning on the AUTSL dataset. Furthermore, this enables us to evaluate which layers contribute to model performance on the AUTSL dataset.

[3]https://huggingface.co/microsoft/xclip-base-patch16-kinetics-600

D. Test Models

Three experiments were conducted to effectively evaluate adaptive fine-tuning for isolated SLR on the AUTSL dataset. All experiments utilised the X-CLIP model pretrained on the Kinetics-600 dataset, explained in Section III-C. The initial experiment focuses on optimising the layer unfreezing step size, the next experiment compares different fine-tuning strategies, and the final experiment re-trains the best model from Experiment 1 with additional data.

- Experiment 1 – Optimising layer step size, s, for adaptive fine-tuning. The focus here is to find the optimal number of layers to unfreeze when validation loss plateaus.
- Experiment 2 – Comparison of three approaches for adapting X-CLIP: adaptive fine-tuning, single-phase fine-tuning, and full transfer learning
- Experiment 3 – Retraining the best model from Experiment 1 with the validation data as additional training data.

V. Results and Discussion

This section elaborates on the results for the various experiments combined with a discussion in Section V-D

A. Experiment 1 – Adaptive Fine-Tuning Step Size

The aim of Experiment 1 is to find the optimal step size s, the number of layers to unfreeze in each adaptive fine-tuning phase. We consider step-sizes ranging from one to three and implement the adaptive fine-tuning strategy detailed in Section III-B. The results for the various step sizes are described in Table I, which shows that the best performing X-CLIP model used a step size of one.

TABLE I: Accuracy Performance by Step Size s on the AUTSL Dataset.

s	Training (%)	Validation (%)	Testing (%)	Epoch
1	99.85	92.05	90.43	150
2	99.52	90.92	89.25	83
3	99.61	90.83	89.01	84

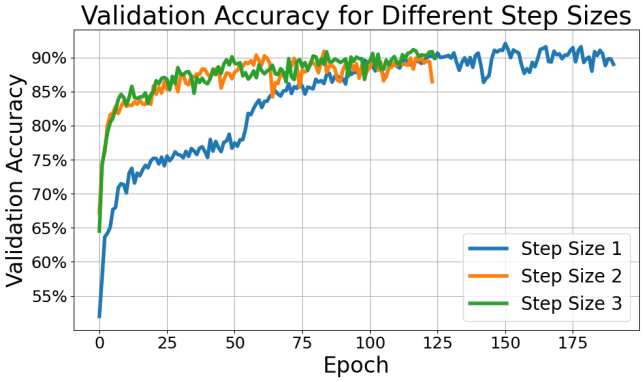


Fig. 4: Validation accuracy during training for different layer step sizes.

The step size of 1 achieved the best overall performance with validation and test accuracies of 92.05% and 90.43%, respectively. This was followed by the step size of 2 with a test accuracy of 89.52%. Highlighting that small layer step sizes result in better performance. However, it is significant to note that models with a larger step size tend to converge faster. Furthermore, the model with a step size of one yielded the lowest validation focal loss score of 0.620 compared to the next best validation loss of 0.666 achieved with a step size of two.

Additionally, the model with a step size of one had unfrozen topmost layers 5-12 at epoch 150, with no further performance gain noted when the final layers, 1-4, were unfrozen. Analysing Figure 4, which shows the validation accuracies during training, the models with higher step sizes learnt rapidly in the first 25 epochs and unlocking more layers past this point resulted in minimal performance gain. Whereas the step size of one had more abrupt changes with increased learning when additional layers were unlocked, as seen just after epoch 53, where layer 11 was unfrozen.

B. Experiment 2 – Comparison of Fine-Tuning Approaches

To effectively compare our adaptive fine-tuning strategy, we take our best model from Experiment 1 and compare it against a single-phase fine-tuning and full transfer learning strategies using the same pretrained X-CLIP model. Using the results from Experiment 1, when applying the single-phase fine-tuning strategy, we unfreeze the 8 topmost layers, which are effectively layers 5-12 of the X-CLIP backbone. The full transfer learning model takes the pretrained X-CLIP model discussed in Section III-C and unfreezes the entire backbone, including the classification head.

TABLE II: Comparison of fine-tuning and transfer learning strategies.

Strategy	Training (%)	Validation (%)	Testing (%)
Single-Phase	99.44	89.18	87.04
Adaptive	99.85	92.05	90.43
Transfer Learning	99.77	89.90	88.29

Table II demonstrates the accuracies achieved across the various fine-tuning strategies. In terms of training accuracy, all strategies produced comparably high scores, with adaptive fine-tuning achieving the highest at 99.85%. The adaptive fine-tuning approach achieved the best overall performance, with validation and testing accuracies of 92.05% and 90.43%, respectively. The single-phase fine-tuning strategy reached a validation accuracy of 89.18% and a test accuracy of 87.04%, while the full transfer learning model obtained a validation accuracy of 89.90% and a test accuracy of 88.29%. Analysing model convergence, the full transfer learning model converged after 51 epochs, significantly quicker than the single-phase and adaptive fine-tuning, which converged after 73 and 100 epochs, respectively.

C. Experiment 3 – Final Model Retraining Using Validation Data

For the final model, we retrain the best model from Experiment 1 using the validation dataset split from the AUTSL dataset as additional training data. The model with a step size of one was selected with pretrained weights loaded from epoch 150 with the lowest validation loss. Minimal training was required, using only 30 epochs with a learning rate of 1×10^{-5} due to the small size of the validation set. The model achieved a final accuracy of 91.10% on the test split of the AUTSL dataset. A 0.67% performance improvement from the training on the training data only.

Across the 226 classes, the model performed well, only struggling with the *Fasting*, *School* and *Sugar* sign class, achieving F-scores of 0.485, 0.500 and 0.533, respectively. The low performance of the *Sugar* sign is expected as it is one of the most underrepresented classes in the dataset. Figure 3 shows three videos of the *Fasting* sign highlighting the significant variation in the manual features as each signer has a unique variation of performing the sign localised to the same region.

D. Discussion

Overall, the X-CLIP model with an adaptive fine-tuning layer step size of one performed well on the isolated signer-independent AUTSL dataset, with an accuracy of 91.10%. The unfreezing of layers with a smaller step size results in significant performance gains largely due to the topmost layers learning the finer features relevant to isolated SLR, and the initial layers keep domain knowledge from the Kinetics-600 dataset, which encompasses human action recognition. This is evident in that the majority of the learning takes place in layers 5-12 of the X-CLIP architecture during fine-tuning.

Furthermore, the adaptive fine-tuning strategy significantly outperformed the single-phase and full transfer learning approaches by 3.39% and 2.14%, respectively. This finding indicates that gradual adaptive fine-tuning tends to preserve pretrained knowledge while supporting improved alignment with the target domain.

It is important to note that all models had overfitting between the train and validation set on the AUTSL dataset. However, this was expected due to the nature of the signer-independence problem and the significant variability between signers. As seen in Experiment 3, a single sign can vary significantly across different signers.

Table III showcases the current state-of-the-art results on the AUTSL dataset. Compared to existing studies, our proposed adaptive fine-tuning strategy, combined with the X-CLIP language image model, performed well on the AUTSL dataset. Outperforming baseline approaches and the Timesformer transformer-based approach. The majority of models that achieved better performance utilised a multi-modal ensemble approach, which requires multi-modal data sources and models combined, which are computationally expensive and do not represent the real-world scenario of SLR, where limited data is available to train models. The impressive performance of the RestNet50-I3D model showcases the strong performance of CNNs' ability to generalise to SLR features. However, the X-CLIP transformer-based model may work better when addressing the continuous SLR problem with longer-term dependencies.

TABLE III: Comparison to existing models on the AUTSL dataset. The first section focuses on unimodal approaches, and the second section encompasses multimodal approaches.

Architecture	Accuracy (%)
CNN+FPM+BLSTM+Attention [11]	47.62
TimeSformer [15]	85.89
SPOTER [15]	84.90
ResNet50-I3D [15]	92.86
VTN+Pose flow [12]	92.92
Neural Ensembler [15]	96.37
SAM-SLR [26]	96.96
X-CLIP Adaptive Fine-Tuning (ours)	**91.10%**

VI. Conclusion

This study demonstrates that adaptive fine-tuning of X-CLIP significantly improves isolated SLR performance on the isolated signer-independent AUTSL dataset, achieving 91.10% accuracy. Through adaptively unfreezing layers, our approach preserves pretrained knowledge from the Kinetics-600 human action recognition dataset while effectively adapting to signer variability and isolated SLR features. The results confirm that small step-size phases of unfreezing yield the best model performance and generalisation ability. Furthermore, adaptive fine-tuning consistently outperformed both single-phase and full transfer learning strategies. Compared to state-of-the-art baselines, our approach proves competitive while requiring fewer computational resources than ensemble methods.

Importantly, this approach advances sustainable AI practices by reducing training overhead and making state-of-the-art SLR more accessible for low-resource African sign languages where limited data is available. Furthermore, this work contributes towards bridging the communication gap between the Deaf and Hard of Hearing communities. Future work will focus on extending adaptive fine-tuning towards benefiting low-resource sign languages using a combination of transfer learning and adaptive fine-tuning. Additional avenues of future exploration include continuous SLR and multi-modal integration.

References

[1] R. Rastgoo, K. Kiani, and S. Escalera, "Sign language recognition: A deep survey," *Expert Systems with Applications*, vol. 164, p. 113794, 2021.

[2] H. Cooper, B. Holt, and R. Bowden, *Sign Language Recognition*. London: Springer London, 2011, pp. 539–562.

[3] R. Rastgoo, K. Kiani, S. Escalera, and M. Sabokrou, "Sign language production: A review," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 3446–3456.

[4] S. Sharma, R. Gupta, and A. Kumar, "Continuous sign language recognition using isolated signs data and deep transfer learning," *Journal of Ambient Intelligence and Humanized Computing*, 2021.



Fig. 5: Three video snapshots of the *Fasting* sign from the AUTSL test split. Across the three videos, significant variation in manual features can be observed for the same sign.

[5] U. Von Agris, J. Zieren, U. Canzler, B. Bauer, and K.-F. Kraiss, "Recent developments in visual sign language recognition," *Universal Access in the Information Society*, vol. 6, pp. 323–362, 2008.

[6] L. Zhu and Y. Yang, "Actbert: Learning global-local video-text representations," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8743–8752.

[7] B. Ni, H. Peng, M. Chen, S. Zhang, G. Meng, J. Fu, S. Xiang, and H. Ling, "Expanding language-image pretrained models for general video recognition," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 1–18.

[8] M. Al-qurishi, T. Khalid, and R. Souissi, "Deep Learning for Sign Language Recognition : Current Techniques , Benchmarks , and Open Issues," *IEEE Access*, vol. 9, pp. 126 917–126 951, 2021.

[9] X. J. Yanqiong Zhang, "Recent advances on deep learning for sign language recognition," *Computer Modeling in Engineering & Sciences*, vol. 139, no. 3, pp. 2399–2450, 2024.

[10] M. Marais, D. Brown, J. Connan, and A. Boby, "An evaluation of hand-based algorithms for sign language recognition," in *2022 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, 2022, pp. 1–6.

[11] O. M. Sincan and H. Y. Keles, "Autsl: A large scale multi-modal turkish sign language dataset and baseline methods," *IEEE Access*, vol. 8, pp. 181 340–181 355, 2020.

[12] M. De Coster, M. Van Herreweghe, and J. Dambre, "Isolated sign recognition from rgb video using pose flow and self-attention," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 3436–3445.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 248–255.

[15] M. Hrúz, I. Gruber, J. Kanis, M. Boháček, M. Hlaváč, and Z. Krňoul, "One model is not enough: Ensembles for isolated sign language recognition," *Sensors*, vol. 22, no. 13, 2022.

[16] N. Sarhan, M. Lauri, and S. Frintrop, "Multi-phase fine-tuning: A new fine-tuning approach for sign language recognition," *KI-Künstliche Intelligenz*, vol. 36, no. 1, pp. 91–98, 2022.

[17] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, p. 9, 2016.

[18] N. Sarhan and S. Frintrop, "Transfer learning for videos: From action recognition to sign language recognition," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 1811–1815.

[19] G. Vrbančič and V. Podgorelec, "Transfer learning with adaptive fine-tuning," *IEEE Access*, vol. 8, pp. 196 197–196 211, 2020.

[20] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8748–8763.

[21] G. Naidu, T. Zuva, and E. M. Sibanda, "A review of evaluation metrics in machine learning algorithms," in *Artificial Intelligence Application in Networks and Systems*, R. Silhavy and P. Silhavy, Eds. Cham: Springer International Publishing, 2023, pp. 15–25.

[22] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, "A short note about kinetics-600," *arXiv preprint arXiv:1808.01340*, 2018.

[23] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijaya-

narasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.
[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.
[25] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019.
[26] S. Jiang, B. Sun, L. Wang, Y. Bai, K. Li, and Y. Fu, "Skeleton aware multi-modal sign language recognition," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 3408–3418, 2021.

Knowledge Distillation Compression of a State-of-the-Art Saliency Object Detection Model

1st Robin Hayman
Smart Sensing and Intelligent Systems
University of Pretoria
Pretoria, South-Africa
robinhayman64@gmail.com

2nd Allan De Freitas
Smart Sensing and Intelligent Systems
University of Pretoria
Pretoria, South-Africa
allan.defreitas@up.ac.za

3rd Herman Myburgh
Smart Sensing and Intelligent Systems
University of Pretoria
Pretoria, South-Africa
herman.myburgh@up.ac.za

***Abstract*—Saliency Object Detection (SOD) is a detection philosophy used in general object detection to detect the most anomalous parts within a frame. Due to the abstract and complex nature of SOD, many Deep neural networks (DNN) models have been developed for SOD. These models display great SOD accuracy and outperform more classic SOD methods but comes at the cost of being large in memory size and requiring a large amount of computational operations.**

This research paper focuses on compressing a state-of-the-art SOD model using the knowledge distillation (KD) DNN compression method. This entails training a smaller version of the same model by using the ground truth and the larger pretrained teacher model. The results in this study show that the student model can potentially surpass the accuracy performance of the larger teacher model, while being significantly smaller and faster.

***Index Terms*—Computer Vision, Saliency Object Detection, Deep Neural Network, Residual Network, Model Compression, Knowledge Distillation, Cross-Modal, Cross-Level**

I. Introduction

One of the key tasks in computer vision is object detection. Object detection is divided into two main categories, namely, specific object detection and general object detection. Specific object detection is used when you know the characteristics or type of object that needs to be detected by the model. However, when there is uncertainty about the type of item, object or entity that has to be detected, a general object detection model is often the only sufficient solution [1], [2].

General object detection models detect objects by generating, what is known as a "saliency map", of the most anomalous or "eye-catching" parts within a captured frame, which should be the objects that the model is attempting to detect. This is known as saliency object detection (SOD) [1], [2].

Classic methods for SOD do exist, but due to the abstract and complex nature of SOD, Deep Neural Network (DNN) models have proven to substantially outperform their classic model counterparts when it comes to SOD accuracy and consistency. This often comes at the price of memory consumption and execution time, due to DNN models having a large amount of parameters and thus requiring many mathematical operations [1], [2].

This work was financially supported by Telkom's Centres of Excellence

In recent times, various methods have been proposed to compress these large DNN models, while retaining most of the SOD accuracy and consistency benefits that come with these DNN SOD models [1], [2]. One such compression method is known as knowledge distillation (KD) [4], [5], [6], which encompasses training a significantly smaller version of a model, using both the ground truth, as well as the larger pretrained model's output. The KD compression DNN method, as well as other compression methods, however, have not really been implemented specifically with DNN-based SOD models.

For this research paper, the KD compression method in SOD models was explored by the implementation of a smaller version of a state-of-the-art DNN SOD model [3]. The smaller version of the model, labelled the "student", was trained using the pretrained larger model, labelled the "teacher", in a classic offline KD configuration [4] - [6]. This means that the loss backpropagated in the student model was contributed by the ground truth, as well as the teacher model.

This paper aims to contribute a study on KD applied to a current state-of-the-art model for SOD tasks. The paper is structured as follows: related topics concerning SOD and KD are first discussed in Section II. The architecture of the chosen SOD model is then described in Section III. The methodology and experimental setup used to obtain the results are presented in Section IV. Finally, the experimental results and their interpretation are discussed in Section V.

II. Related Works

Many of the state-of-the-art SOD models [3] [7], [8], [9] and [10] either use a fully convolutional network-based (FCN-based) multi-data stream, FCN-based multi-operation, FCN-based top-down/bottom-up architecture or even a hybrid between the previously mentioned architectures. This is mostly an attempt to extract more localised detail and/or to retain the finer details that are lost as the frame data passes through multiple CNN layers.

Single data stream DNN models consisting of only a RGB-image [7], [8], [9], as well as multi-data stream DNN models consisting of a RGB-image and a depth/infrared map [3], [10], are still being used and researched. Multi-data stream DNN models tend to deliver better SOD accuracy due to the

extra information; however, it comes at the real-world cost of requiring a set of sensors for the model.

DNN KD is the process of transferring already learned knowledge, which has been learned during the training phase, to a new DNN. The simplest and most common method of KD is to use an already trained network as the teacher network, to guide the loss values that are backpropagated to the student network [4] – [6].

One of the most common use cases of KD is to train a smaller version of the teacher DNN. This is often accompanied by the learner's DNN weights of the remaining layers, being initialized to the teacher's DNN weights [4] – [6]. A diagram of this approach can be seen in Figure 1:

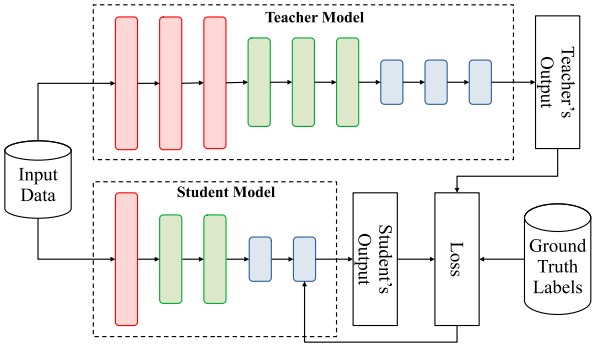


Fig. 1: Diagram that demonstrates how KD works on a basic level

III. Proposed Models

The research paper closely implements the cross-modal and cross-level fusion learning framework model [3]. The model uses two residual networks (ResNets) as its backbone for the initial feature extraction of the RGB image and the depth map.

It then further implements novel attention enhancement modules (AEM) and modal feature fusion modules (MFM), which process the output feature map of each of the main Resnet blocks. The AEM modules draw attention to channels with a high concentration of activations and to spatial locations within the feature maps with a high concentration of activations across channels. The MFM modules fuse the RGB and depth feature maps into a single feature map. This process is enhanced with the aid of the output feature map from the next higher level MFM module [3].

Lastly, a novel Graph Reasoning Module (GRM) was implemented for the three highest-level MFM output feature maps in order to extract localised high-level and low-level features using graph convolution techniques. The first and second level feature maps was also used by every GRM module in order to balance the low-level and high-level features when constructing the vertex projection matrix. The final saliency map was extracted from the three GRM output feature maps using a decoder module (DM) [3]. A high-level diagram of the CMCL model can be seen in Figure 2.

A. ResNet Backbone

Two conventional ResNets [11] were implemented for the backbone of the CMCL model, one for extracting features from the RGB image and one for extracting features from the depth map. Two ResNet50s with bottleneck residual blocks [11] were implemented for the teacher model, while two smaller ResNet18s with basic residual blocks [11] were implemented for the student model.

B. Attention Enhancement Module

The RGB-image feature map consists of features like colour, appearance, and texture, while the depth-image feature map consists of features like distance, shape and spatial relationships. The level at which each of these feature types was extracted varies between channels of a particular feature map. The purpose of the AEM module is to enhance the features of the RGB and depth feature maps that are important for the particular scene while suppressing the other unimportant features [3].

The AEM module begins taking an RGB-image feature map (FR_i) and a depth-image feature map (FD_i), which were generated by the respective ResNet50 backbones, as input. The two feature maps are each passed into their respective channel attention block (CAB), to generate channel attentive feature maps (see Section III-B1 for more info). The mathematical formula for the previously mentioned process can be expressed as,

$$\begin{aligned} CR_i &= \text{CAB}(FR_i), \\ CD_i &= \text{CAB}(FD_i), \end{aligned} \tag{1}$$

where CR and CD are the channel attentive feature maps for the RGB image and depth image, respectively, and i is the ResNet convolutional level index. After the channel attentive feature maps are generated, both maps are passed to two spatial attention blocks (SAB). For the first SAB block, the RGB-image channel attentive feature map is set as "input x" and the depth-image channel attentive feature map is set as "input y" (see Section III-B2 for more info), while the input settings are vice versa for the second SAB block [3]. The mathematical formula for the previously mentioned process can be expressed as,

$$\begin{aligned} AR_i &= \text{SAB}(CR_i, CD_i), \\ AD_i &= \text{SAB}(CD_i, CR_i), \end{aligned} \tag{2}$$

where AR and AD are the spatial attentive feature maps for the RGB-image and depth-image, respectively.

1) Channel Attention Block: A feature map has multiple channels since each channel consists of a different combination of feature types of varying degrees that were extracted from the image space. The CAB block aims to enhance the attention of the important channels (channels with levels of activation) for a feature map while suppressing the non-important channels [3].

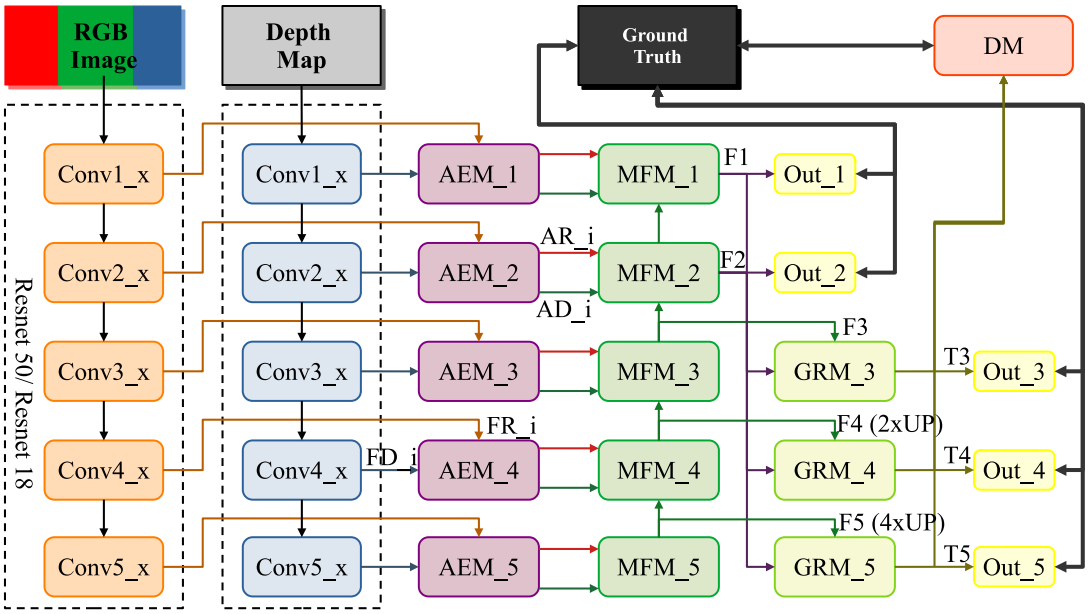


Fig. 2: High-level diagram of the CMCL model.

The CAB block begins by first applying global max pooling (GMP) and global average pooling (GAP) to the input feature map (X). "Global" means that the pooling kernel's width and height are the same as the feature map's width and height. This effectively means that the maximum value (GMP) and average value (GAP), per channel, for the feature map are calculated. The global map max and average pooling features are then each passed through their own respective 1×1 convolutional layer (for more info see Section III-F1).

The purpose of the convolutional layers is to extract features and patterns between channels on a global map scale. It should also be noted that no channel scaling occurs during the convolutional layers, meaning that the number of output channels is equal to the number of input channels for the feature map.

After convolution, the global map max pooling and global map average pooling features are combined in an element-wise summation fashion. The features are then scaled between 0 and 1 with a Sigmoid function to effectively obtain the global map feature set.

Finally, the global map feature set is then fused with the input feature map. This is done in an element-wise multiplication fashion along the input feature map's height and width axis since the global map pooling feature set is only a single value per channel [3]. The mathematical formula for the previously mentioned process can be expressed as,

$$\begin{aligned} \mathrm{CAB}(x) &= x \otimes_{Spatial} w_{global}, \\ w_{global} &= \mathrm{Sigmoid}(\mathrm{M}(x) \oplus \mathrm{A}(x)), \\ \mathrm{M}(x) &= \mathrm{Conv}_{1\times1}^{Max}(\mathrm{GMP}(x)), \\ \mathrm{A}(x) &= \mathrm{Conv}_{1\times1}^{Avg}(\mathrm{GAP}(x)), \end{aligned} \tag{3}$$

where w_{global} is the global map pooling feature, $\otimes_{Spatial}$ is an element-wise multiplication along the width and height axis of the feature map, $\oplus$ is element-wise summation, GMP() and GAP() is the global max and average pooling functions, $Conv_{1\times1}()$ is a convolutional layer with a kernel size of 1×1 and Sigmoid() is the sigmoid function. A diagram of the CAB block can be seen in Figure 3:

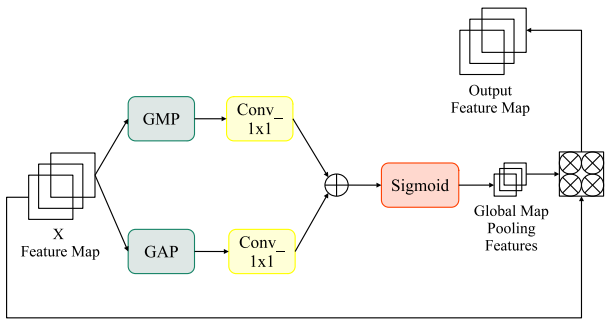


Fig. 3: Low-level diagram of the CAB block

2) Spatial Attention Block: While the CAB block aims to enhance the attention of channel features on a global spatial scale, the SAB block, in contrast, aims to enhance the attention for features at certain spatial locations, with high overall channel activation [3].

The SAB block begins by first fusing the features of the two input feature maps in an element-wise summation fashion. Channel max pooling (CMP) and channel average pooling (CAP) are then applied to the summed feature map. Channel pooling is the operation of calculating the maximum (CMP)

channel value or average channel (CAP) value for every pixel within the feature map. The output of a channel pooling operation is thus a feature map with the same dimensions as the input feature map but with only a single channel containing the maximum/average along the channel axis.

The two resulting channel pooling feature maps are then concatenated along the channel axis, resulting in one feature map with two channels. The feature map is then passed through a 3×3 convolutional layer that not only extracts features and patterns from the channel-pooled feature map but also scales the channels down from two to a single channel again. The resulting feature map is then passed through a Sigmoid function to scale the values in the feature map between 0 and 1 and thus obtain the global channel pooling feature set.

Finally, the global channel pooling feature set is then fused with the input feature map. This is done in an element-wise multiplication fashion along the input feature map's channel axis since the global channel pooling feature set only has a single channel [3]. The mathematical formula for the previously described process can be expressed as,

$$\begin{aligned} \text{SAB}(x, y) &= x \otimes_{Channel} z_{global}, \\ z_{global} &= \text{Sigmoid}(\text{Conv}_{3\times3}(\text{C}(x, y)), \\ \text{C}(x, y) &= \text{Concat}(\text{CMP}(x \oplus y), \text{CAP}(x \oplus y)), \end{aligned} \tag{4}$$

where x is the main feature map, y is the secondary feature map, z_{global} is the global channel pooling feature-set, Sigmoid() is the sigmoid function, $\text{Conv}_{3\times3}()$ is a convolutional layer with a 3×3 kernel size, Concat() is the concatenation function along the channel axis, CMP() is the channel max pooling function, CAP() is the channel average pooling function, $\oplus$ is element-wise summation and $\otimes_{Channel}$ is element-wise multiplication along the channel axis. A diagram of the SAB block can be seen in Figure 4:

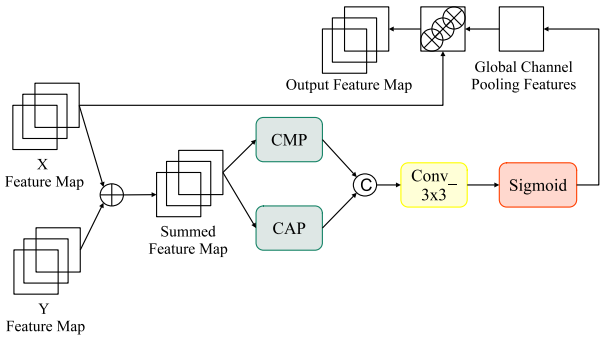


Fig. 4: Low-level diagram of the SAB block.

C. Modal Feature Fusion Module

The main purpose of the MFM module is to effectively combine and fuse the RGB-image feature map and the depth-image feature map. To allow for effective fusion, a strategy known as cross-modal fusion is deployed. This method fuses the features of the current feature map with a feature map of one of the following layers, as these feature maps have higher-level features. Inspiration for this method was drawn from another SOD model called the UNet [3], [13].

The MFM module has three inputs, which are the spatial attentive feature maps for the RGB-image (AR_i) and Depth-image (AD_i) and the MFM modal feature map of the following MFM module. In the case of the last ($i = 5$) MFM module that will not have an MFM modal feature map from the following MFM module, only the attentive feature maps are used as input.

Firstly, the following MFM modal feature map is spatially scaled with an upscaling layer and channel scaling is applied with a 1×1 convolutional layer. These scaling operations bring the following MFM feature map's dimensions in line with the attentive feature map's dimensions, ensuring that element-wise operations can occur.

The scaled following MFM modal feature map is then element-wise multiplied with both attentive feature maps to fuse its higher-level features with the attentive maps [3]. The mathematical formula for the previously mentioned operations can be expressed as,

$$\begin{aligned} F_{i+1}^{Scale} &= \text{Conv}_{1\times1}(\text{UP}(F_{i+1})), \\ m_i &= AR_i \otimes F_{i+1}^{Scale}, \\ n_i &= AD_i \otimes F_{i+1}^{Scale}, \end{aligned} \tag{5}$$

where $\otimes$ is element-wise multiplication, UP() is the upscaling layer that performs bi-linear upscaling and F_{i+1} is the following MFM modal feature map. The resulting element-wise feature maps (m_i and n_i) are then element-wise summed as well as element-wise multiplied, effectively fusing the RGB-image and Depth-image features into one feature map.

The resulting element-wise summed map and element-wise multiplied map are then concatenated along the channel axis, effectively creating a single feature map with double the amount of channels when compared to the input attentive feature maps. The concatenated feature map is then finally passed through a 3×3 convolutional layer, which has the purpose of extracting features and patterns between pixels and channels, while also scaling the number of channels of the concatenated feature map down to the same amount as the input attentive feature maps for element-wise operations [3]. The mathematical formula for the previously mentioned operations can be expressed as,

$$k_i = \text{Conv}_{3\times3}(\text{Concat}((m_i \oplus n_i), (m_i \otimes n_i))) \tag{6}$$

where k_i is the final RGB-depth fused feature map. Finally, to get the current MFM modal feature map (F_i), the scaled following MFM modal feature map is element-wise summed with the final RGB-depth fused feature map. This has the same feature retention effect as the residual blocks of the ResNet backbone. The mathematical formula for this can be expressed as,

$$F_i = k_i \oplus F_{i+1}^{Scale}, \tag{7}$$

where F_i is the current MFM modal feature map. For the final MFM module that will not have a following MFM modal feature map, all operations that involved the following MFM modal feature map were stripped out. The mathematical formula for the final MFM modal module can thus be condensed down to the formula in Equation 8:

$$F_5 = \text{Conv}_{3\times3}(\text{Concat}((AR_5 \oplus AD_5), (AR_5 \otimes AD_5))), \tag{8}$$

where F_5 is the final MFM modal feature map and AR_5 and AD_5 are the final spatial attentive feature maps from the AEM module [3]. A diagram of the MFM module with the cross-level enhancement can be seen in Figure 5 and without the cross-level enhancement can be seen in Figure 6:

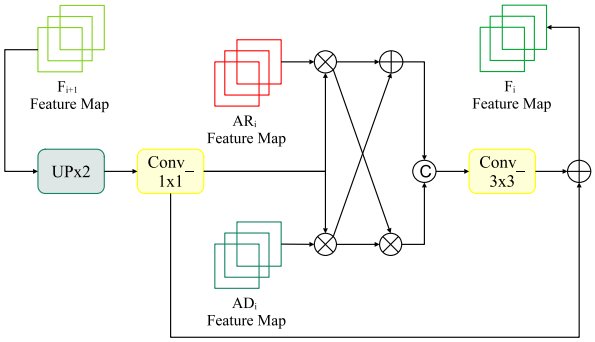


Fig. 5: Low-level diagram of the MFM module with cross-level enhancement.

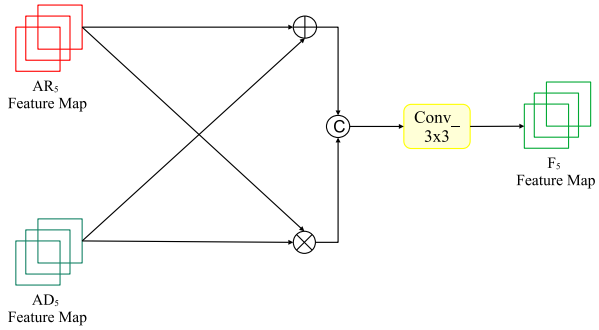


Fig. 6: Low-level diagram of the MFM module without cross-level enhancement.

D. Graph Reasoning Module

The purpose of the GRM module is to extract more non-localised features while also performing cross-modal operations between the low-level extracted features and high-level extracted features. This is done by utilising the low-level modal feature maps (F_1 and F_2) and high-level modal feature maps (F_3, F_4 or F_5) to generate a graph structure, which in turn passes through a graph convolutional network. The result of the GRM module is graph-optimised feature maps that have enhanced low-level features and less noise [3].

The GRM module has three feature map inputs, two of which are the low-level modal feature maps (F_1 and F_2), while the other input is one of the three high-level feature maps (F_3, F_4 or F_5). All of the high-level feature maps are upsampled beforehand to match the height and width dimensions of the F_3 feature map, before being passed to the GRM module as input. The upsampling layer for F_4 and F_5 uses bilinear upsampling.

The GRM layer begins by first applying channel scaling on the input high-level modal map. The channels of the high-level modal map are scaled down to the GRM channel constant (L) amount, which, for the implemented CMCL model, is 32 channels. Channel scaling is applied to the high-level modal map by passing it through a 1×1 convolutional layer. However, since the high-level modal feature map is used for encoded feature input to the GCN layer, as well as to create the map-to-graph projection matrix, two independent 1×1 convolutional layers are used for the channel scaling operation [3]. The mathematical formula for the previously mentioned operations can be expressed as,

$$\begin{aligned} U &= \text{Conv}^{U}_{1\times1}(F_{3/4/5}), \\ V &= \text{Conv}^{V}_{1\times1}(F_{3/4/5}), \end{aligned} \tag{9}$$

where $F_{3/4/5}$ is the high-level modal feature map, V is encoded input features for the GCN layer and U is the high-level feature map used for the map-to-graph projection matrix. Next, the two low-level feature maps are channel downscaled to the GRM channel constant by using a 1×1 convolutional layer and spatial downscaled by using an average pooling layer. The mathematical formula for the previously mentioned operations can be expressed as,

$$\begin{aligned} W_1 &= \text{Down}(\text{Conv}^{W_1}_{1\times1}(F_1)), \\ W_2 &= \text{Down}(\text{Conv}^{W_2}_{1\times1}(F_2)), \end{aligned} \tag{10}$$

where F_1 and F_2 are the low level modal feature maps, Down() is the downscale function and W_1 and W_2 is the GRM channel scaled low-level modal feature maps. Next, the high-level features and low-level features are fused into a single high-low feature map. This is done by element-wise multiplying the high-level feature map (U) with the two low-level feature maps W_1 and W_2 before the resulting multiplied feature maps are element-wise summed [3].

The resulting feature map is then averaged-pooled with a kernel size and stride based on the pixel-vertex ratio chosen. This means that the average pooling operation downscales the feature map to the number of vertices that the graph structure in the GCN layer will have. The mathematical equation for the previously described process can be expressed as,

$$Z = \text{AP}((U \otimes W_1) \oplus (U \otimes W_2)), \tag{11}$$

where AP() is the average pooling operation and Z is the vertex encoded feature map. Thereafter, the vertex-encoded

feature map and the high-level feature map (U) are first transformed into matrices by unfolding both of them around the channel axis, before matrix-multiplying the vertex-encoded feature map with the transposed high-level feature map. The resulting matrix is then passed through a softmax layer along the vertex axis, to obtain the projection matrix (P_{matrix}) [3].

Softmaxing along the vertex axis means that the weights towards all of the vertices, for every pixel, will sum to 1 [3]. The mathematical equation for the previously described process can be expressed as,

$$P_{Matrix} = \text{Softmax}(Z_{Matrix} \times U^{T}_{Matrix}), \tag{12}$$

where Softmax() is the softmax function. After the projection matrix has been calculated, the high-level feature map with encoded features (V) is projected onto the graph structure before applying graph convolution. This is done by first unfolding the high-level feature map along the channel axis and then matrix multiplying it with the projection matrix [3]. The mathematical equation for the previously described process can be expressed as,

$$G_{Matrix} = \text{GCN}(P_{Matrix} \times V_{Matrix}), \tag{13}$$

where GCN() is the graph convolutional layer (see Section III-F2) and G_{Matrix} is the graph convoluted feature matrix. After the convoluted feature matrix is calculated, the matrix is re-projected back to a pixel feature matrix (E_{Matrix}) by multiplying it with the transposed projection matrix. The pixel feature matrix is then transformed back to a feature map (E) by folding it to the same dimensions as the original feature map [3].

After the feature map is obtained, its channels are upscaled back to the original $F3$, $F4$ or $F5$ amount with a 1×1 convolutional layer. Lastly, the original $F3/F4/F5$ map is element-wise summed with the resulting map, for residual feature forth-bringing, to finally obtain the optimised GRM map ($T3/T4/T5$) [3]. The mathematical equation for the previously described process can be expressed as,

$$\begin{aligned} E_{matrix} &= P^{T}_{Matrix} \times G_{Matrix}, \\ T_{3,4,5} &= F_{3,4,5} \oplus \text{Conv}_{1\times1}(E). \end{aligned} \tag{14}$$

A high-level and low-level diagram of the GRM module can be seen in Figure 7 and Figure 8, respectively:

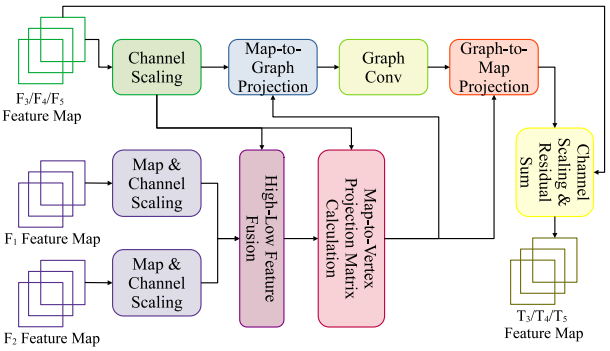


Fig. 7: High-level diagram of the GRM module.

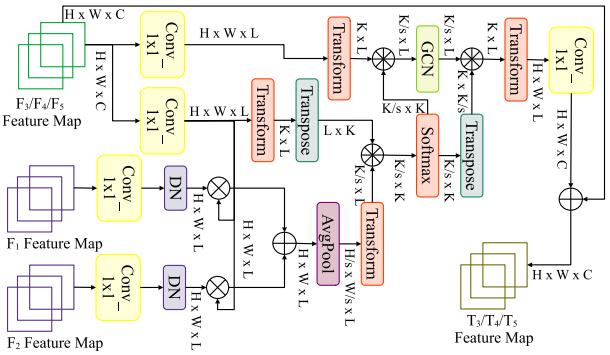


Fig. 8: Low-level diagram of the GRM module. The feature map dimensions can be seen along the flow arrows.

E. Decoding Module

The DM module's main objective is to generate the prediction saliency map. The DM module begins by first taking the three optimised GRM maps (T_3, T_4 and T_5) as input. The DM module first concatenates the three feature maps along the channel axis into one large feature map. The channel amount for the resulting concatenated feature map is thus the sum of the channels of the three optimised GRM maps [3].

After concatenation, the resulting feature map is passed through a 3×3 convolutional layer. The purpose of the convolutional layer is to extract features and patterns for the final time between pixels and between channels (and feature maps due to the concatenation). No channel scaling occurs during the 3×3 convolutional layer, meaning that the output channels and input channels are the same amount [3].

Finally, the convoluted feature map is passed through the output layer (see Section III-F3) to obtain the final saliency prediction map [3]. The mathematical formula for the previously mentioned operations can be expressed as,

$$S_{pred} = \text{Out}(\text{Conv}_{3\times3}(\text{Concat}(T_3, T_4, T_5))), \tag{15}$$

where Out() is the output layer and S_{pred} is the final saliency prediction map. A diagram of the DM module can be seen in Figure 9:

F. Complex Layers

1) Convolutional Layers: Two types of convolutional layers are used in the AEM, MFM, GRM and DM module. These convolutional layers are the $\text{Conv}_{1\times1}()$ and $\text{Conv}_{3\times3}()$ convolutional layer. The $\text{Conv}_{1\times1}()$ convolutional layer uses a kernel with a size of 1×1 with a stride of 1. To achieve a "Same" padding mode, no padding is included in the feature map for the $\text{Conv}_{1\times1}()$ convolutional layer [3].

The $\text{Conv}_{3\times3}()$ convolutional layer uses a kernel with a size of 3×3 with a stride of 1. To achieve a "Same" padding mode,

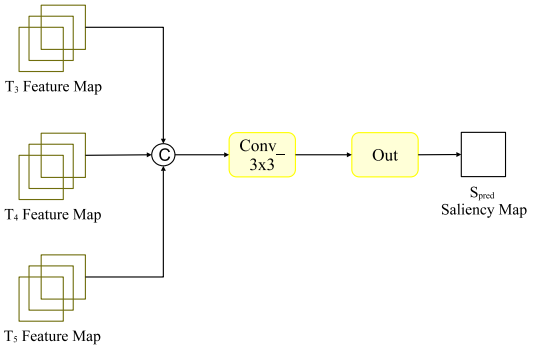


Fig. 9: Low-level diagram of the decode module.

one pixel row and column of zero padding are included in the feature map for the $\text{Conv}_{3\times3}()$ convolutional layer.

Both convolutional layer types make use of batch normalisation with learnable affine parameters after convolution, and both convolutional layer types implement an activation function after the batch normalisation layer. Due to the learnable batch normalisation affine parameters that already have a bias, no learnable bias parameter was included during convolution [3].

2) GCN Layer: Traditionally, the GCN layer takes in two inputs, an adjacency matrix that describes how the edges interconnect with the vertices within the graph and a vertex-embedded feature matrix that describes the feature values for each vertex [14], [15]. However, for the image processing procedure that the layer is based on [16], the adjacency matrix can be obtained by simply matrix multiplying the vertex-embedded feature matrix with a transposed version of itself [3], [16].

After the adjacency matrix has been calculated, it is simply multiplied by the vertex-embedded feature matrix and the learnable weight matrix. No self-loops are added to the adjacency matrix, and no normalisation is done with the degree matrix. Thereafter, the resultant matrix is batch normalised before being passed through an activation function [16]. The mathematical equation for the previously described process can be expressed as,

$$\begin{aligned} Y &= \epsilon(\text{BN}(A \times X \times W_g)), \\ A &= X \times X^T \end{aligned} \tag{16}$$

where X is the vertex embedded feature matrix, A is the adjacency matrix, W_g is the learnable weight matrix BN() is the batch normalization function, $\epsilon()$ is the activation function and Y is the convoluted vertex embedded feature matrix [16].

3) Output Layers: The output layer consists of a convolutional layer with a kernel size of 1×1, a kernel stride of 1 and no padding and an upsampling layer with bilinear scaling. Unlike the conventional convolutional layers, the output layer's convolutional layer does have bias parameters, due to not having a batch normalisation layer [3].

The main purpose of the output layer is to channel downscale the input feature map down to only a single channel since saliency maps are effectively grey-scale images. The purpose of the upsample layer is to upsample the feature map to the same dimensions as the input RGB-image and depth-image, to match the dimensions of the ground truth saliency map for evaluation [3].

G. Teacher and Student Model Differences

The original CMCL model [3] uses the convolutional part of the ResNet50 as a backbone and was the same ResNet variant used for the teacher model in this research paper. For the student model, the ResNet50 variant backbone was instead replaced by the smallest ResNet18 variant [11]. Further, the original model [3] uses the ReLU function for activation functions. The implemented teacher model, however uses the GELU function instead, in order to avoid the "Dead ReLU Problem" [12] for maximum learning performance. The student model uses ReLU6 for its activation function and replaces the sigmoid functions with hard sigmoid functions in order to make the model more integer-only arithmetic friendly for quantisation (see Section VI).

Other than the previously mentioned changes, the teacher and student models are the same architecturally as [3].

IV. Methodology

A. Experimental Setup

First, the teacher SOD model described in Section III was implemented in PyTorch and trained using the COME15K dataset. The dataset comprises over fifteen thousand samples, each consisting of an RGB image, a depth map, and a ground truth saliency map.

The RGB image and depth map are used as inputs to the model, while the ground truth is used during backpropagation to train it. During the training phase, random data augmentation – including image flipping, rotation, hue adjustments, brightness changes, and saturation variations – was applied to the input data to prevent overfitting.

The model was trained using a custom loss function. The custom loss function implemented on a high level is simply the loss values summed by the previously mentioned six supervision points. The high-level mathematical formula for the custom loss function can thus be defined in Equation 17 as,

$$L_{Tot} = L_{Pre} + L_{Low} + L_{Opt}, \tag{17}$$

where L_{Tot} is the total loss that will be back-propagated, L_{Pre} is the loss calculated for the predicted saliency map (S), L_{Low} is the loss calculated for the low-level modal feature maps generated by the first 2 MFM modules (F_1 and F_2) and L_{Opt} is the loss calculated for the graph optimized feature maps generated by the three GRM modules (T_3, T_4 and T_5).

L_{Pre}, L_{Low} and L_{Opt} losses are all calculated by summing the binary cross entropy loss and dice loss, between the particular feature map and the ground truth saliency map,

within the previously mentioned feature map category. All feature maps are accompanied by an output layer beforehand in order to upscale the feature to the correct dimensions and reduce the channel amount to a single channel.

After the teacher model was trained, the student model was implemented and trained using the KD DNN compression method [4] - [6] with the same COME15K dataset and data augmentation strategy. For the student model, a ground truth loss and a teacher loss were calculated, with both losses being weighted equally during backpropagation. The ground truth loss was calculated the same as the teacher model, while the teacher loss was calculated by estimating the loss between each supervised feature map (F_1, F_2, T_3, T_4, T_5 and S) for the teacher and the student model. The binary cross-entropy loss and dice loss function were also used for the teacher loss.

After both models were implemented and trained, seven test datasets were used to evaluate their accuracy. The seven test datasets are DES, LFSD, NJU2K, NLPR, SIP, SSD, and STERE. Each dataset tests different salient object detection scenarios, containing either single or multiple salient objects within a scene. Although the seven test datasets vary in the number of samples they include, they follow the same sample format as the COME15K dataset.

Further, the accuracy of the models was also evaluated when one of the three novel modules (AEM, MFM and GRM) of the model was removed, as this indicates how much the particular module contributes to the accuracy of the model. The AEM and GRM modules could be easily removed from the models; however, since RGB and depth features are merged in the MFM module, only the cross-level enhancement component was removed from all MFM modules, effectively changing the architecture of all five MFM modules to the diagram in Figure 6. The memory and computational requirements of both models were also measured.

B. Evaluation Metrics

To evaluate the overall accuracy of the generated SOD maps for a scene, multiple SOD metrics are used. This is due to the complexity of SOD maps.

Three of the main SOD performance metrics used are the mean average error (MAE) [2], the F-measure (F-score) [2] and the E-measure (E-score) [17].

1) MAE: The MAE SOD performance metric is the easiest performance metric to calculate, as well as grasp. However, ease of calculation comes at the cost of the metric not considering local structural similarities but rather just the similarity on a global scale. Further, MAE calculates the error of a map on continuous values (normalised between 0 and 1), meaning that the generated SOD map is not binarised [2].

The MAE SOD performance metric calculates the average error that the generated saliency map (S) makes, compared to the ground truth saliency map (GT), on a per-pixel basis. The mathematical formula for MAE can thus be defined as,

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x,y) - GT(x,y)|, \tag{18}$$

where W and H are the width and height of the saliency maps, respectively. MAE is a simple and decent metric to get an overall idea, on a global scale, of the accuracy of the predicted saliency maps [2].

2) F-score: The F-score metric focuses on the SOD accuracy when the SOD model has to make a final decision on what pixels of the saliency map are salient and what pixels of the saliency map are background. This entails binarising the generated output saliency map.

The generated binarized saliency map (M) and ground truth map (GT) is then used to calculate the F-score F_{beta} using the formula in Equation 19:

$$F_\beta = \frac{(1+\beta^2) \times Pre \times Rec}{(\beta^2 \times Pre) + Rec}, \tag{19}$$

where Pre is the precision value between M and GT, Rec is the recall value between M and GT and β is the F-score weight. For SOD measurements $\beta = 0.3$, to weight precision more, as the recall is deemed less important than precision [3].

3) E-score: The E-score measurement was developed as a means of more precisely measuring the similarity between a binarised generated saliency map and the ground truth, as the F-score is deemed a bit rudimentary [17].

On a high level, the E-score formula can be summed up in Equation 20 as,

$$E_{score} = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} \Phi(x,y), \tag{20}$$

where H and W are the height and width of the generated and ground truth map, respectively, and Φ is the enhanced alignment matrix. The enhanced alignment matrix is calculated using the mathematical formulas,

$$\begin{aligned} \Phi(x,y) &= f(\zeta(x,y)), \\ f(x) &= \frac{1}{4}(1+x)^2, \end{aligned} \tag{21}$$

where ζ is the alignment matrix and $f(x)$ is the convex function. The alignment matrix is calculated using the formula,

$$\zeta = \frac{2 \times (\eta_{GT} \otimes \eta_{FM})}{(\eta_{GT} \otimes \eta_{GT}) \oplus (\eta_{FM} \otimes \eta_{FM})}, \tag{22}$$

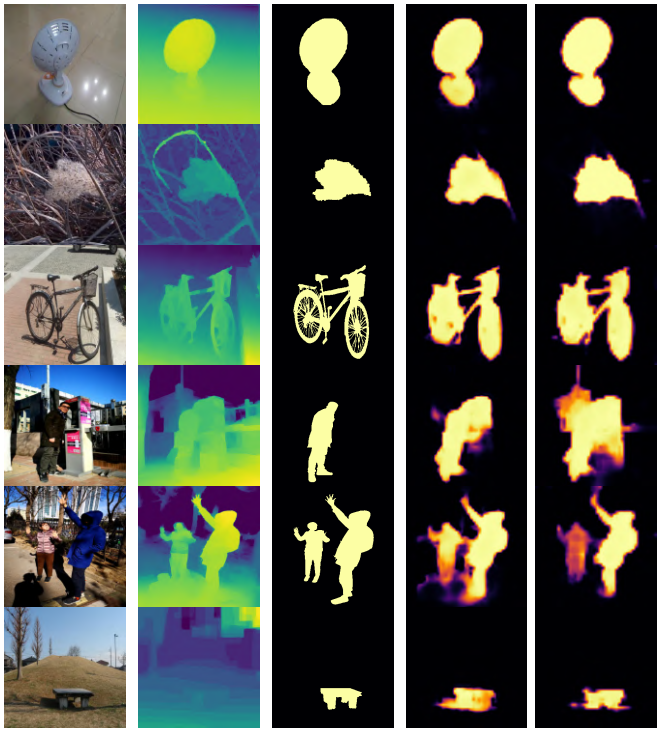
where η are the bias matrices of the binarised generated saliency map (FM) and the ground truth (GT), $\otimes$ is element-wise multiplication and $\oplus$ is an element-wise summation. The formula for calculating the bias matrix is defined as,

$$\eta_I = I - (\mu_I \times A), \tag{23}$$

where I is a binary SOD map, μ is the binary SOD maps mean and A is a matrix with the same dimensions as I and has only 1 values [17].

V. Experimental Results and Discussion

Some samples of the testing datasets can be seen in Figure 10, where it displays the RGB image, depth image, the ground truth, the teacher's generated SOD map and the student's generated SOD map.



(a) RGB (b) Depth (c) GT (d) Teacher (e) Student

Fig. 10: Visual sample results

A. CMCL Model Accuracy Results

The accuracy results for the teacher and student CMCL model can be seen in Table I. The table displays the loss (explained in sub-section IV-A), MAE, F-score and E-score (explained in sub-section IV-B) achieved by the CMCL teacher model for the seven testing datasets. The results for the full model are shown as well as when one of three novel module types is removed (explained in sub-section IV-A).

From Table I, it can be seen that the teacher model was able to achieve an average loss, MAE, F-score and E-score of 4.264, 0.103, 0.691, and 0.807, respectively, across the seven testing datasets. The student model was able to achieve a loss, MAE, F-score and E-score of 4.062, 0.098, 0.693 and 0.817, respectively, across the seven testing datasets. This not only matches the accuracy results of the teacher model but even slightly surpasses it on all metrics. Further, it can also be seen that this better accuracy performance was consistent across all of the datasets, with the teacher only being more accurate in a few metrics of a few datasets.

When the AEM modules were removed, it was seen that the SOD accuracy dropped between 2% and 10% for the teacher and student most of the time, depending on the metric and dataset. However, there were singular cases (like the loss metric with the SIP dataset) where accuracy was increased by removing the AEM modules. However, on average, the AEM was still a net positive for the CMCL module. When comparing the accuracy between the teacher and student model for the AEM module removed, the most accurate model was much more inconsistent, with the teacher being the superior model in some cases, while the student being the superior model in other cases. This indicates that the AEM module was contributing slightly more to the student model's accuracy than the teacher model.

When the cross-level enhancement for the MFM modules was removed, the drop in accuracy was much more severe than with the AEM module, with it being between 20% and 150% depending on the metric and dataset. This indicates that the MFM module was largely contributing to the accuracy of the CMCL model, which makes sense, since this is also the module where the fusion between RGB and depth features happens. When comparing the accuracy between the teacher and student model for the MFM module removed, the most accurate model was always the teacher model. This indicates that the cross-level enhancement was contributing to the accuracy even more in the student model than the teacher model.

When the GRM modules were removed, it was seen that the SOD accuracy dropped between 4% and 8% for the teacher while dropping between 20% and 30% for the student, depending on the metric and dataset. This substantially larger drop difference for the student indicates that the GRM modules are contributing much more to the accuracy in the smaller student model than the larger teacher model. This is also apparent when comparing accuracy between the teacher and student models for the GRM modules removed, since the teacher model performs better in every metric for every dataset.

B. CMCL Model Memory and Computational Results

Pytorch info was used to capture memory and computational information about both the teacher and student models. The important information is displayed in Table II.

From the memory and computational results, it can clearly be seen that the student model is only about 20% the size of the teacher model and also takes only 37% of the time that the teacher model will take per frame while achieving slightly better performance.

VI. Conclusion

A smaller student variant of the CMCL model was successfully implemented and trained using the KD compression method, where the student model was trained using the ground truth and the larger teacher model from [3]. The student model was able to perform overall slightly better than the teacher model while being significantly smaller, less computationally expensive and faster.

Further, from the module contribution analysis, it was found that the AEM was contributing the least amount to the accuracy of the models, with there even being singular cases where the models were performing better without the AEM module,

TABLE I: SOD accuracy results for the teacher and student CMCL model. The arrows in the metric column indicate whether higher or lower values are better. Brackets in the difference column indicate better results for the student model. *The average was calculated over the seven datasets.

Dataset	Metrics	Full Model			AEM Removed			MFM Removed			GRM Removed		
		Teacher	Student	Diff	Teacher	Student	Diff	Teacher	Student	Diff	Teacher	Student	Diff
DES	Loss ↓	2.704	2.772	0.068	3.187	3.577	0.390	7.331	9.028	1.697	3.241	3.714	0.473
	MAE ↓	0.063	0.054	(0.009)	0.082	0.091	0.009	0.193	0.229	0.036	0.072	0.078	0.006
	F-score ↑	0.683	0.697	(0.014)	0.639	0.628	0.011	0.374	0.337	0.037	0.649	0.630	0.019
	E-score ↑	0.840	0.852	(0.012)	0.800	0.791	0.009	0.589	0.545	0.044	0.816	0.821	(0.005)
LFSD	Loss ↓	4.640	3.874	(0.766)	4.314	3.898	(0.416)	8.472	10.58	2.108	4.662	5.121	0.459
	MAE ↓	0.119	0.113	(0.006)	0.115	0.114	(0.001)	0.207	0.231	0.024	0.125	0.140	0.015
	F-score ↑	0.779	0.772	(0.007)	0.773	0.752	0.021	0.608	0.569	0.039	0.765	0.720	0.045
	E-score ↑	0.828	0.841	(0.013)	0.836	0.837	(0.001)	0.707	0.675	0.032	0.824	0.816	0.008
NJU2K	Loss ↓	4.271	3.784	(0.487)	4.616	4.083	(0.533)	8.282	9.900	1.618	4.489	4.856	0.376
	MAE ↓	0.105	0.098	(0.007)	0.116	0.111	(0.005)	0.207	0.229	0.022	0.112	0.125	0.013
	F-score ↑	0.740	0.739	(0.001)	0.709	0.705	0.004	0.530	0.501	0.029	0.721	0.683	0.038
	E-score ↑	0.823	0.840	(0.017)	0.806	0.815	(0.009)	0.672	0.501	0.171	0.816	0.808	0.008
NLPR	Loss ↓	4.806	5.260	0.454	5.217	5.666	0.449	7.787	9.694	1.907	5.101	5.549	0.448
	MAE ↓	0.109	0.107	(0.002)	0.122	0.132	0.010	0.208	0.244	0.036	0.116	0.130	0.014
	F-score ↑	0.531	0.548	(0.017)	0.493	0.495	(0.002)	0.319	0.295	0.024	0.505	0.481	0.024
	E-score ↑	0.725	0.734	(0.009)	0.692	0.692	0.000	0.536	0.505	0.031	0.703	0.694	0.009
SIP	Loss ↓	3.964	3.637	(0.327)	3.672	3.401	(0.271)	7.764	8.528	0.764	4.341	4.603	0.262
	MAE ↓	0.098	0.090	(0.008)	0.096	0.092	(0.004)	0.193	0.226	0.033	0.106	0.117	0.011
	F-score ↑	0.729	0.739	(0.010)	0.712	0.715	(0.003)	0.518	0.471	0.047	0.703	0.675	0.028
	E-score ↑	0.833	0.848	(0.015)	0.830	0.839	(0.009)	0.680	0.628	0.052	0.817	0.815	0.002
SSD	Loss ↓	5.253	5.182	(0.071)	6.064	5.656	(0.408)	8.307	9.756	1.449	5.431	5.827	0.396
	MAE ↓	0.122	0.124	0.002	0.145	0.146	0.001	0.219	0.235	0.016	0.132	0.150	0.018
	F-score ↑	0.676	0.668	0.008	0.622	0.616	0.006	0.481	0.472	0.009	0.651	0.610	0.041
	E-score ↑	0.785	0.780	0.005	0.739	0.739	0.000	0.627	0.617	0.010	0.766	0.748	0.018
STERRE	Loss ↓	4.209	3.931	(0.278)	4.112	4.027	(0.085)	8.184	9.710	1.526	4.446	5.133	0.687
	MAE ↓	0.106	0.103	(0.003)	0.108	0.112	0.004	0.209	0.239	0.030	0.114	0.131	0.017
	F-score ↑	0.701	0.690	0.011	0.693	0.667	0.026	0.500	0.462	0.038	0.683	0.632	0.051
	E-score ↑	0.818	0.822	(0.004)	0.814	0.803	0.009	0.665	0.619	0.046	0.806	0.786	0.020
Average*	Loss ↓	4.264	4.062	(0.202)	4.455	4.330	(0.125)	8.018	9.599	1.581	4.530	4.972	0.442
	MAE ↓	0.103	0.098	(0.005)	0.112	0.114	0.002	0.205	0.230	0.025	0.111	0.124	0.013
	F-score ↑	0.691	0.693	(0.002)	0.663	0.654	0.018	0.476	0.444	0.032	0.668	0.633	0.035
	E-score ↑	0.807	0.817	(0.010)	0.788	0.788	0.000	0.639	0.604	0.035	0.793	0.784	0.009

TABLE II: Memory and computational performance of the teacher and student model.

Metric	Teacher	Student
Total Parameters	196.3 M	34.18 M
Total Multiplication and Addition Operations	509.6 G	113.0 G
Activation forward pass size	9.302 GB	2.137 GB
Parameter size	785.1 MB	136.7 MB
Real-world time per frame	0.830 s	0.305 s
CPU time per frame	13.28 s	4.870 s

while the cross-level enhancement of the MFM module was contributing the most to the accuracy of the CMCL model.

Lastly, from the module contribution analysis, it was found that the GRM was contributing significantly more to the accuracy of the smaller student model when compared to the larger teacher model. This could indicate that graph convolution modules could be a particularly worthy addition to models that are aiming to be small, while retaining good SOD accuracy.

Future work could focus on the addition of another DNN compression technique, like model quantisation, to include in the smaller student model to further reduce the size of the model and make the model mostly integer operations, which is less computationally complex.

ACKNOWLEDGMENT

I would like to acknowledge the unwavering support provided by my supervisors, the Department of Electrical, Electronic and Computer Engineering at the University of Pretoria, as well as the Smart Sensing and Intelligent Research Group. Lastly, I would like to acknowledge the financial support of Telkom's Centre for Connected Intelligence (CCI).

REFERENCES

[1] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, "Salient object detection in the deep learning era: An in-depth survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 6, pp. 3239–3259, 2022.

[2] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," IEEE Transactions on Image Processing, vol. 24, no. 12, pp. 5706–5722, 2015.

[3] Y. Peng, Z. Zhai, and M. Feng, "RGB-D salient object detection based on cross-modal and cross-level feature fusion," IEEE Access, vol. 12, pp. 45 134–45 146, 2024.

[4] C.-S. Lin, C.-N. Tsai, J.-S. Jwo, C.-H. Lee, and X. Wang, "Heterogeneous student knowledge distillation from bert using a lightweight ensemble framework," IEEE Access, vol. 12, pp. 33 079– 33 088, 2024.

[5] L. Beyer, X. Zhai, A. Royer, L. Markeeva, R. Anil, and A. Kolesnikov, "Knowledge distillation: A good teacher is patient and consistent," 2022. [Online]. Available: https://arxiv.org/abs/2106.05237

[6] W. Li, J. Wang, T. Ren, F. Li, J. Zhang, and Z. Wu, "Learning accurate, speedy, lightweight cnns via instance-specific multi-teacher knowledge distillation for distracted driver posture identification," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 10, pp. 17 922–17 935, 2022.

[7] H. Li, “Pyramid spatial context features for salient object detection,” IEEE Access, vol. 8, pp. 88 518–88 526, 2020.
[8] J. Li, J. Su, C. Xia, M. Ma, and Y. Tian, “Salient object detection with purificatory mechanism and structural similarity loss,” IEEE Transactions on Image Processing, vol. 30, pp. 6855–6868, 2021.
[9] Y.-H. Wu, Y. Liu, L. Zhang, M.-M. Cheng, and B. Ren, “Edn: Salient object detection via extremely-downsampled network,” IEEE Transactions on Image Processing, vol. 31, pp. 3125– 3136, 2022.
[10] F. Xiao, Z. Pu, J. Chen, and X. Gao, “Dgfnet: Depth-guided cross-modality fusion network for rgb-d salient object detection,” IEEE Transactions on Multimedia, vol. 26, pp. 2648–2658, 2024.
[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 06 2016, pp. 770–778.
[12] L. L. Lu Lu, Y. S. Yeonjong Shin, Y. S. Yanhui Su, and G. E. K. George Em Karniadakis, “Dying relu and initialization: Theory and numerical examples,” Communications in Computational Physics, vol. 28, no. 5, p. 1671–1706, Jan. 2020. [Online]. Available: http://dx.doi.org/10.4208/cicp.OA-2020-0165
[13] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.
[14] S. Zhang, H. Tong, J. Xu, and Maciejewski, “Graph convolutional networks: a comprehensive review,” vol. 6, 11 2019.
[15] F. Wu, T. Zhang, A. H. de Souza Jr. au2, C. Fifty, T. Yu, and K. Q. Weinberger, “Simplifying graph convolutional networks,” 2019. [Online]. Available: https://arxiv.org/abs/1902.07153
[16] Y. Li and A. Gupta, ‘Beyond Grids: Learning Graph Representations for Visual Recognition’, in Advances in Neural Information Processing Systems, 2018, vol. 31.
[17] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, ‘Enhanced-alignment Measure for Binary Foreground Map Evaluation’, in Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, July 2018, pp. 698–704.

Evaluating Model Compression Techniques for Efficient Edge AI Deployment

Mondli L Zungu
Department of Computer Science
University of Zululand
KwaDlangezwa, South Africa
mondlilizwi52@gmail.com

Skhumbuzo G Zwane
NextGen Enterprises and Institutions
CSIR
Pretoria, South Africa
SZwane7@csir.co.za

Matthew O Adigun
Department of Computer Science
University of Zululand
KwaDlangezwa, South Africa
AdigunM@unizulu.ac.za

Abstract*— Deploying artificial intelligence models on edge devices presents challenges due to limited computational power, memory, and energy. Model compression is essential to reduce the size and computational requirements of AI models, enabling their deployment on smartphones, IoT sensors, and other edge devices. This paper evaluates three major compression techniques which are quantization, pruning, and tensor decomposition, applied to the YOLOv8n model for object detection. The study compares their impact on accuracy, model size, inference speed, and energy efficiency. Findings indicate that quantization achieves the best balance by reducing size and improving inference speed with minimal accuracy loss, pruning yields high energy savings but at the cost of accuracy, and tensor decomposition provides a balanced trade-off. The research underscores the importance of compression in enabling real-time Edge AI applications and highlights the quantization model compression technique as the most suitable technique for efficient edge deployment.

Keywords*— Edge AI, Model Compression, Pruning, Quantization, Tensor Decomposition, YOLOv8n

I. INTRODUCTION

Artificial intelligence (AI) has emerged as a transformative force across sectors such as healthcare, transportation, agriculture, and smart cities. Its ability to process vast amounts of data using deep learning has enabled applications ranging from medical image analysis to real-time object detection and natural language processing [1]. Central to many of these applications are deep neural networks (DNNs), which, despite their accuracy and robustness, are computationally expensive and memory intensive [2].

The emergence of Edge AI addresses the limitations of cloud-based systems by enabling local processing on devices such as smartphones, drones, and Internet of Things (IoT) sensors [3]. By processing data closer to its source, Edge AI reduces latency, minimizes bandwidth usage, and strengthens data privacy. These advantages make it particularly attractive for real-time systems such as autonomous driving, smart surveillance, and medical diagnostics. However, the deployment of large-scale AI models on edge devices remains challenging because of hardware limitations in terms of computational power, memory, and energy [4].

Object detection models such as YOLO (You Only Look Once) have become widely recognized for balancing accuracy with speed in real-time applications [5]. Nevertheless, newer versions such as YOLOv8 contain millions of parameters, making them difficult to deploy efficiently on edge hardware. Running such uncompressed models on low-power devices often results in higher latency, energy inefficiency, and reduced reliability [2].

To address this problem, model compression techniques have been developed. These methods aim to reduce the size and computational requirements of models without severely degrading their performance. Among them, the most widely adopted are quantization, pruning, and tensor decomposition.

Quantization reduces weight and activation precision, leading to smaller models and faster inference [6][7]. Pruning removes unimportant weights or filters, thereby reducing the number of operations [8][9]. Tensor decomposition approximates large tensors with lower-rank structures to preserve performance while lowering complexity [10][11].

However, previous research have indicated that these approaches may present trade-offs between accuracy, inference speed, and energy efficiency. For instance, while pruning can reduce operations, aggressive pruning can lead to degraded performance in complex detection tasks [12][4]. Similarly, tensor decomposition has shown promise in maintaining accuracy but is computationally expensive to implement in certain contexts [13].

Therefore, given the increasing demand for AI deployment in resource-constrained environments, it becomes necessary to evaluate these compression techniques systematically to determine the most effective technique for compressing edge AI models. This study focuses on applying quantization, pruning, and tensor decomposition to the YOLOv8 object detection model, with the goal of identifying the most suitable approach for Edge AI deployment. The findings aim to provide actionable insights into balancing performance with efficiency, enabling practical real-time deployment of AI in diverse domains.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

II. Related Work

The deployment of AI models in edge environments has stimulated extensive research into model compression techniques. The need arises from the inherent resource constraints of edge devices, which include limited memory, processing power, and energy capacity [4]. Several methods have been proposed to address these challenges, with quantization, pruning, and tensor decomposition emerging as the most widely explored.

Quantization has been recognized as an effective means of reducing model size by lowering the numerical precision of weights and activations. Han, et. al [2] demonstrated that reducing precision from 32-bit floating-point to 8-bit integers can reduce model size by up to four times. Jacob et. al [6] further introduced quantization-aware training, which helps maintain accuracy by simulating quantized weights during the training process. Choudhary et. al [7] noted that quantization is particularly advantageous for Edge AI because it not only reduces memory footprint but also lowers power consumption, with some studies [7] [14] reporting up to 70% energy savings.

Pruning has been another major focus of research. Early work by Han et. al [8] on "deep compression" showed that pruning redundant connections, followed by quantization and coding, could reduce model size by up to 90% with minimal loss of accuracy. Molchanov et. al [9] extended this work by developing pruning methods based on sensitivity analysis, showing significant reductions in floating-point operations (FLOPs) . While unstructured pruning creates sparse networks that are less hardware-efficient, structured pruning which removes entire filters or channels, has been shown to achieve more practical improvements on edge hardware [12].

Tensor decomposition approaches have been explored to optimize high-dimensional weight matrices. Jaderberg et. al [13] applied Canonical Polyadic (CP) decomposition to convolutional layers, achieving up to 50% reduction in computational cost while retaining 95% accuracy. Similarly, Kim et. al [11] demonstrated that Tucker decomposition could compress convolutional neural networks without significant accuracy loss, particularly benefiting mobile applications . In addition, Lebedev et. al [10] confirmed that tensor decomposition can reduce the number of parameters substantially while retaining model performance, although implementation complexity remains a barrier.

Beyond these individual techniques, hybrid approaches have gained traction. Han et. al [8] combined pruning and quantization, demonstrating that their joint application yields models that are both smaller and faster. Hinton et. al [15] introduced knowledge distillation, training smaller "student" networks to replicate larger "teacher" models, further improving efficiency. Another hybrid approach [16] proposed combining knowledge distillation with quantization to produce highly compact models without substantial accuracy loss.

Despite these advances, challenges remain. Quantization can lead to degraded accuracy in high-precision tasks [17], pruning may oversimplify models when applied aggressively [18], and tensor decomposition introduces computational overhead during decomposition [19]. Furthermore, hybrid approaches require careful tuning, and their effectiveness often depends on the hardware environment [20]. These limitations point to the need for systematic evaluation of these compression methods on specific architectures, such as YOLOv8, to identify the most effective technique for promoting real-time, high performance edge AI implementations.

III. Methodology

This study employed an experimental research design to evaluate the impact of three compression techniques which are quantization, pruning, and tensor decomposition on the YOLOv8n object detection model.

A. Research Design

The study adopted a quantitative experimental approach. The baseline YOLOv8n model was trained and evaluated on a benchmark dataset, after which each compression technique was applied independently. The performance of the compressed models was compared against the baseline to determine trade-offs between accuracy, model size, inference speed, and energy consumption, as illustrated in Figure 1.

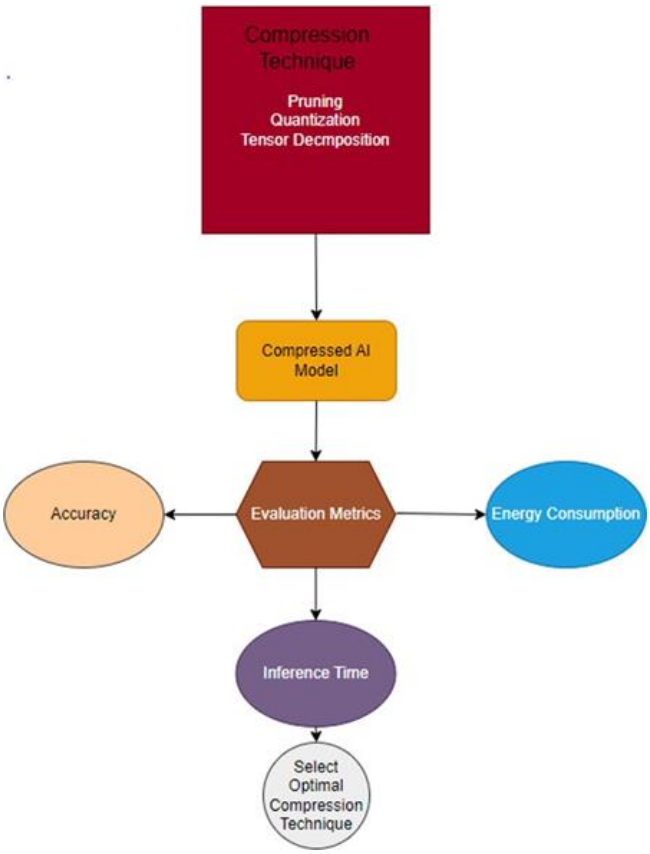


Figure 1 Study Design

B. Dataset

The experiments utilized dataset sourced from Kaggle, consisting of approximately 200,000 images across multiple classes and annotated for object detection tasks. The dataset was diverse in terms of background, scale, and illumination, ensuring robustness in evaluation. To enhance generalization, data

augmentation techniques such as flipping, rotation, scaling, and color jittering were applied.

C. Compression Techniques

Three state-of-the-art compression techniques were implemented:

Quantization: The precision of model weights was reduced from 32-bit floating-point (FP32) to 8-bit integer (INT8) representations. Both post-training quantization (PTQ) and quantization-aware training (QAT) were utilized. PTQ applies numerical scaling to approximate model weights after training, while QAT simulates low-precision effects during training to preserve model accuracy in the quantized format.

Quantization can be mathematically described as:

$$W_q = round(\frac{w}{s}) \times s \qquad (1)$$

where wq is the quantized weight, w the original weight, and s the scaling factor.

Pruning: Both structured and unstructured pruning strategies were implemented. Structured pruning involves removing entire channels or filters, whereas unstructured pruning eliminates individual weights. The pruning ratio P is defined as:

$$P = \frac{n_{pruned}}{n_{total}} \times 100\% \qquad (2)$$

where $npruned$ is the number of parameters removed, and $ntotal$ is the original number of parameters

Tensor Decomposition: Canonical Polyadic (CP) and Tucker decomposition methods were applied to factorize convolutional layers. A convolutional kernel $W \in R^{K\times K\times C}in\times Cout$ was decomposed into smaller tensors, thereby reducing computational complexity. For CP decomposition, the kernel can be expressed as:

$$W = \sum_{r=1}^{R} a_r \otimes b_r \otimes c_r \qquad (3)$$

where R is the rank of decomposition, and $\otimes$ denotes the outer product.

IV. Experiments and Results

A. Evaluation Metrics

The performance of the baseline and compressed YOLOv8n models was assessed using four key metrics. Accuracy was measured using mean Average Precision (mAP), which averages the precision values across all object classes and provides a standard benchmark in object detection tasks. Model size was recorded in megabytes before and after compression, with percentage reduction calculated to quantify the efficiency of each technique. Inference speed was evaluated in frames per second (FPS) by measuring the number of images processed per second, providing an indication of real-time applicability. Finally, energy consumption was estimated by monitoring GPU power usage during inference, where total energy was calculated as the product of average power and inference time. These metrics were selected to provide a balanced evaluation of accuracy, efficiency, and practicality for deployment on resource- constrained edge devices.

Accuracy – was measured using mean Average Precision (mAP).

$$mAP = \frac{1}{N} \sum_{i-1}^{N} AP_i \qquad (4)$$

Where API is the average precision of class i, and N is the number of object classes.

Model Size – Measured as the total storage required (in MB) before and after compression

Inference Speed – Measured in frames per second (FPS):

$$FPS = \frac{N_{frames}}{T_{inference}} \qquad (5)$$

where $Nframes$ is the number of frames processed, and

$Tinference$ is the total inference time.

Energy Consumption – Computed by monitoring power usage during inference

$$E = P \times t \qquad (6)$$

where E is the energy consumed (in Joules), P the average power usage (Watts), and t the inference time (seconds).

B. Experimental Setup and Execution

The experiments were conducted on Google Colab Pro+ with access to NVIDIA Tesla T4 and A100 GPUs, supported by Intel Xeon CPUs, 25 GB of RAM, and 100 GB of storage. The software environment included Python 3.10, PyTorch 2.0, and the Ultralytics YOLOv8 framework, with additional libraries such as NumPy, Torchvision, and OpenCV for preprocessing and evaluation.

The dataset consisted of approximately 200,000 annotated images resized to 640 × 640 pixels. Data augmentation techniques, including rotation, scaling, and flipping, were applied to improve generalization, and the dataset was divided into 70% training, 20% validation, and 10% testing sets.

The YOLOv8n baseline model was trained for 100 epochs with a batch size of 16, a learning rate of 0.01, and Stochastic Gradient Descent as the optimiser. After baseline training,

quantization (PTQ and QAT), pruning (structured and unstructured with ratios between 20% and 70%), and tensor decomposition (CP and Tucker) were applied. Performance was evaluated using mean Average Precision (mAP) for accuracy, model size in megabytes, inference speed in frames per second, and energy consumption estimated via GPU power usage. This setup ensured fair and consistent comparison of compression techniques against the baseline model.

C. Experimental results

The experimental evaluation began with establishing the baseline performance of the YOLOv8n model on the prepared dataset. Table 1 presents results obtained from the conducted experiments, while Figure 3 and 4 visualize the model sizes, overall accuracy and inference speed and energy consumptions, respectfully. As shown in Table 1, the baseline achieved strong detection accuracy, with a high mean Average Precision (mAP), but required substantial storage space and computational resources. Its inference speed was limited, and energy consumption was relatively high.

Table 1 Comparison of Compression Techniques

Model / Technique	Model Size (MB)	Overall Accuracy (%)	Inference Speed	Energy Consumption
Baseline (Uncompressed YOLOv8)	12.5 MB	98.6%	Slow (High latency)	High (Consumes more power)
Pruning	5.3 MB	76.12%	Fastest inference	Moderate energy usage
Quantization	4.8 MB	95.07%	Fast (slightly slower than Pruning)	Efficient (moderate)
Tensor Decomposition	3.5 MB	79.22%	Moderate speed	Most energy efficient

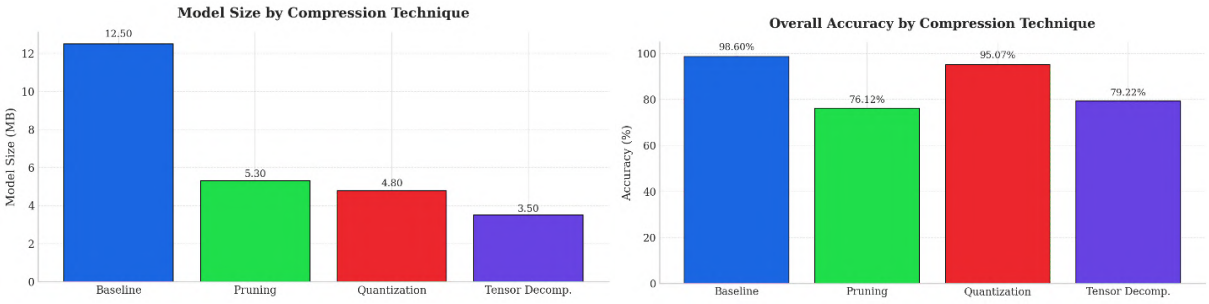


Figure 3 Graphs Showing the Model Sizes and Accuracy

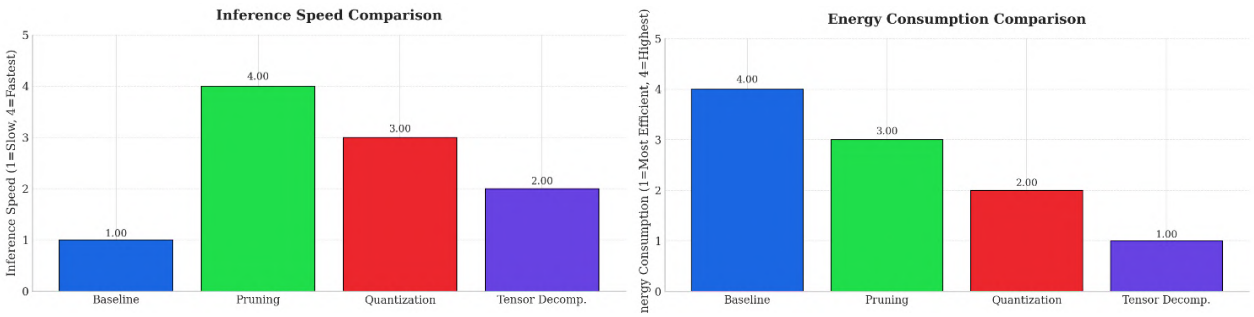


Figure 4 Graphs showing time consumed and Inference speed

These baseline results highlighted the challenges of deploying uncompressed deep learning models on edge devices and provided the benchmark against which all compression methods were compared.

Applying quantization to the baseline model yielded significant improvements. The model size was reduced by almost three-quarters, dropping from the original size of 13.5 MB to 5.3 MB a fraction suitable for mobile and IoT devices. Despite this reduction, quantized models retained accuracy levels very close to the baseline, with only a 1–2% decrease in mAP, illustrated in Figure 3. Inference speed increased considerably, confirming quantization's ability to improve responsiveness. Energy consumption also declined due to lower computational demands, demonstrating the suitability of quantization for real-time edge deployment (Figure 4).

Pruning, on the other hand, produced varied results depending on the pruning ratio. At moderate pruning levels (20–40%), the model size decreased substantially, and energy consumption fell by over 50%, with a modest reduction in inference time.

However, as the pruning ratio increased beyond 50%, accuracy dropped sharply, making heavily pruned models unsuitable for high-precision applications. This confirms pruning's value when power efficiency is prioritised, though its benefits come at the expense of predictive accuracy.

The results of tensor decomposition fell between quantization and pruning. Decomposition reduced model size and computational cost while maintaining accuracy closer to the baseline compared to heavily pruned models. However, the implementation of CP and Tucker decomposition introduced additional computational steps, which affected efficiency gains. Although decomposition does not outperform quantization in terms of overall balance, it demonstrated promise for use cases where preserving accuracy is more important than maximizing compression ratios. The results also highlight the potential for hybrid approaches, where techniques such as pruning and quantization could be combined to achieve greater efficiency while reducing the limitations of any single method.

D. Result Analysis and Discussions

The findings illustrate the inherent trade-offs in applying model compression techniques for Edge AI deployment. Quantization emerged as the most balanced approach, achieving substantial reductions in model size and improvements in inference speed while maintaining accuracy close to the baseline. This outcome demonstrates its practicality for real-time applications such as mobile health diagnostics, smart surveillance, and IoT-based monitoring systems, where efficiency must be achieved without sacrificing predictive reliability. These results support earlier work by Jacob , who identified quantization as one of the most effective techniques for deploying deep learning models on constrained devices [6][7].

In contrast, pruning demonstrated its strength in reducing computational demand and energy consumption. By removing redundant parameters, pruning enabled models to achieve up to 50% energy savings, which is valuable for battery-powered edge devices such as drones or remote sensors. However, the experiments also revealed that higher pruning ratios resulted in a significant decline in accuracy, confirming the sensitivity of pruning to over-parameter reduction. This observation aligns with prior studies by Han et. al [2] and Molchanov et. al [9] which reported that excessive pruning compromises robustness and generalization, especially in complex object detection tasks.

Tensor decomposition provided an intermediate solution by offering moderate reductions in size and computational complexity while preserving accuracy better than heavily pruned models. However, decomposition methods such as CP and Tucker introduced additional implementation complexity and computation during decomposition, which may limit their adoption in highly resource-constrained hardware. Nonetheless, these results suggest that tensor decomposition could be beneficial for specialized use cases, such as embedded AI systems in industrial automation, where maintaining accuracy is critical and the hardware can tolerate additional computational steps.

Taken together, the analysis underscores that the choice of compression method must be application specific. In this study, quantization provides the best balance between efficiency and accuracy for real-time object detection, making it the most reliable option for widespread Edge AI deployment. Pruning remains suitable when energy efficiency is the overriding priority, while tensor decomposition serves as a compromise for applications requiring stability in accuracy despite limited compression gains. The results also highlight the need for hybrid approaches, where techniques such as pruning and quantization could be combined to achieve greater efficiency while reducing the limitations of any single method.

V. CONCLUSION

The purpose of this study was to evaluate and compare model compression techniques to enable the deployment of advanced object detection models, specifically YOLOv8n, on edge devices with limited computational and energy resources. The results confirmed that compression is indispensable in balancing the trade-off between accuracy, efficiency, and practicality in resource-constrained environments. Among the tested approaches, quantization demonstrated the most consistent and reliable performance, achieving reductions of up to seventy-five percent in model size and delivering faster inference with minimal accuracy loss. Pruning provided the greatest energy savings, making it highly relevant for scenarios where battery life is the primary concern, although its tendency to degrade accuracy at higher pruning levels limits its broader application. Tensor decomposition was shown to maintain accuracy more effectively than pruning but introduced additional

implementation challenges that must be addressed before it can be widely adopted.

The findings contribute to the growing body of knowledge on Edge AI by providing a comparative analysis of compression methods on a modern detection framework. This study confirms that compression cannot be applied in isolation but must be matched to the needs of specific applications. For instance, quantization is ideal for real-time object detection tasks such as traffic monitoring or mobile health diagnostics, while pruning is better suited for low-power devices that priorities efficiency over precision. Tensor decomposition, though less practical at present, offers opportunities for specialized domains where maintaining accuracy is critical, and hardware resources are less restricted.

This research also provides practical implications for industry adoption. By demonstrating that compression can enable state-of-the-art models to operate effectively on edge hardware, the study highlights a pathway for integrating AI into diverse applications, from autonomous systems to smart city technologies. Importantly, the study underscores that model compression is not a one-size-fits-all solution but rather a set of strategies that should be carefully adapted to the context of use.

Future research should explore hybrid compression techniques that combine quantization, pruning, and tensor decomposition to leverage their complementary strengths. Hardware-aware optimization strategies should also be developed to align compression techniques with the specific capabilities of target devices, ensuring maximum efficiency. Additionally, evaluating compressed models under real-world deployment scenarios, such as on embedded IoT boards or mobile devices, will provide deeper insights into their performance. Through these directions, the deployment of efficient, accurate, and scalable AI models on edge devices can be significantly advanced, supporting innovation across healthcare, transportation, and other critical sectors.

ACKNOWLEDGMENT

I would like to acknowledge the unwavering support provided by my supervisors, the Department of Computer Science at the University of Zululand as well as the center for mobile services at the University of Zululand.

REFERENCES

[1] Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2020). *Understanding deep learning requires rethinking generalization. Communications of the ACM*, 63(3), 107–115. https://doi.org/10.1145/3373710

[2] Han, S., Mao, H., & Dally, W. J. (2016). *Deep compression: Compressing deep neural networks with pruning, trained quantization, and Huffman coding. International Conference on Learning Representations (ICLR).* https://arxiv.org/abs/1510.00149

[3] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). *Edge computing: Vision and challenges. IEEE Internet of Things Journal*, 3(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

[4] Liu, Z., Sun, M., Zhou, T., Huang, G., & Darrell, T. (2017). *Learning efficient convolutional networks through network slimming. Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2755–2763. https://doi.org/10.1109/ICCV.2017.299

[5] Redmon, J., & Farhadi, A. (2018). *YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.* https://doi.org/10.48550/arXiv.1804.02767

[6] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., & Kalenichenko, D. (2018). *Quantization and training of neural networks for efficient integer-arithmetic-only inference. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2704–2713. https://doi.org/10.1109/CVPR.2018.00286

[7] Choudhary, T., Mishra, V., Goswami, A., Sarangapani, J., & Rao, C. (2020). *A comprehensive survey on model compression and acceleration. Artificial Intelligence Review*, 53(7), 5113–5155. https://doi.org/10.1007/s10462-020-09816-8

[8] Han, S., Pool, J., Tran, J., & Dally, W. (2015). *Learning both weights and connections for efficient neural network. Advances in Neural Information Processing Systems (NeurIPS).*

[9] Molchanov, P., Tyree, S., Karras, T., Aila, T., & Kautz, J. (2017). *Pruning convolutional neural networks for resource efficient inference. International Conference on Learning Representations (ICLR).*

[10] Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I., & Lempitsky, V. (2014). *Speeding-up convolutional neural networks using fine-tuned CP-decomposition. International Conference on Learning Representations (ICLR).*

[11] Kim, Y. D., Park, E., Yoo, S., Choi, T., Yang, L., & Shin, D. (2016). *Compression of deep convolutional neural networks for fast and low power mobile applications. International Conference on Learning Representations (ICLR).*

[12] Wen, W., Wu, C., Wang, Y., Chen, Y., & Li, H. (2016). *Learning structured sparsity in deep neural networks. Advances in Neural Information Processing Systems (NeurIPS).*

[13] Jaderberg, M., Vedaldi, A., & Zisserman, A. (2014). *Speeding up convolutional neural networks with low rank expansions. British Machine Vision Conference (BMVC).* https://doi.org/10.5244/C.28.88

[14] Deng, L., Li, G., Han, S., Shi, L., & Xie, Y. (2020). *Model compression and hardware acceleration for neural networks: A comprehensive survey. Proceedings of the IEEE*, 108(4), 485–532. https://doi.org/10.1109/JPROC.2020.2976475

[15] Hinton, G., Vinyals, O., & Dean, J. (2015). *Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.* https://doi.org/10.48550/arXiv.1503.02531

[16] Polino, A., Pascanu, R., & Alistarh, D. (2018). *Model Compression via Distillation and Quantization. arXiv preprint arXiv:1802.05668.* Available at: https://arxiv.org/abs/1802.05668 (Accessed: 12 November 2025).

[17] Gholami, A. et al. (2021). *A survey of quantization methods for efficient neural network inference. arXiv preprint arXiv:2103.13630.* Available at: https://arxiv.org/abs/2103.13630 (Accessed: 12 November 2025).

[18] Blalock, D., Ginsburg, B., Guttag, J., & Shen, H. (2020). *What is the state of neural network pruning? arXiv preprint arXiv:2003.03033.* Available at: https://arxiv.org/abs/2003.03033 (Accessed: 12 November 2025).

[19] Kossaifi, J. et al. (2023). *Tensor decomposition for model reduction in neural networks: A review. arXiv preprint arXiv:2304.13539.* Available at: https://arxiv.org/abs/2304.13539 (Accessed: 12 November 2025).

[20] Guan, Q. et al. (2017). *A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282.* Available at: https://arxiv.org/abs/1710.09282 (Accessed: 12 November 2025).

Improving Sleep: An Analysis of AI-driven Sleep Monitoring Capabilities in Consumer-grade Smartwatches

Tadiwanashe Musemesi
Department of Computing Sciences
Nelson Mandela University
Gqeberha, South Africa
Tadiwanashe.Musemesi@mandela.ac.za

Brenda Scholtz
Department of Computing Sciences
Nelson Mandela University
Gqeberha, South Africa
Brenda.Scholtz@mandela.ac.za

Simone Beets
Department of Computing Sciences
Nelson Mandela University
Gqeberha, South Africa
Simone.Beets@mandela.ac.za

***Abstract*—Artificial Intelligence (AI) has the potential to transform sleep health and reduce the burden of non-communicable diseases in South Africa. However, there is a lack of studies that provide an understanding of AI-driven sleep monitoring capabilities of smartwatches and the sensors that enable them. This study analyses and compares the AI capabilities and related sensor requirements of popular consumer smartwatches adopted by South Africans. The research methods used were a non-systematic literature review, a document analysis, and a comparative analysis. Seven AI capabilities were identified from the literature review, namely sleep staging, sleep quality scoring, disorder risk screening, anomaly detection, coaching/feedback, chronotype classification, and recovery assessments. Sleep staging and sleep quality scoring were commonly adopted by all of the reviewed smartwatch brands while the remaining capabilities were implemented differently among the different brands. Additionally, the sensor integration capabilities varied between the brands, with a higher number of sensors integrated into Apple and Samsung smartwatches, while Garmin places its focus on performance-oriented metrics. This paper offers a conceptual framework for comparing AI capabilities in consumer-grade wearables, including smartwatches, with the goal of supporting South African validation priorities and guiding future research.**

Keywords—smartwatch, artificial intelligence, sensors, sleep monitoring

I. INTRODUCTION

Non-communicable diseases (NCDs), such as cardiovascular disease, diabetes mellitus type 2, and hypertension, are an emergent public health issue on the African continent, as well as emerging in South Africa [1]. NCDs are now significant contributors to morbidity and death both globally and in South Africa [2], [3]. South Africa reports higher rates of obesity, diabetes, and hypertension than many other regions globally [4]. These NCDs are closely associated with sleep disorders and poor sleep behaviours [2], [5]. Evidence of these associations supports integrated health strategies that prioritise early identification, self-monitoring, and behaviour-focused interventions.

Research studies that focus on improving sleep are therefore aligned with Sustainable Development Goal 3 (SDG 3), which focuses on aiming for the positive health and well-being of all individuals at all ages. One of the goals of SDG 3 involves reducing premature deaths from NCDs through prevention, treatment, and improving mental health [6]. Sleep has generally been recognised as a fundamental health determiner and modifiable factor for reducing NCD risk [7].

Healthy sleep has increasingly been recognised as a public health priority for promoting cognitive performance, emotional stability, metabolic control, cardiovascular health, and overall well-being [8]. Sleep monitoring and improvement thus relates to the health of individuals and global health targets.

South African studies now show alarming sleep quality trends across different population groups [9]–[11]. For instance, one nation-wide study established that 9.1% of respondents older than 50 years had problems with sleep, while another study found that nearly one-third of older respondents from rural areas had suboptimal sleep [9], [10]. A 2023 longitudinal study of rural adults aged 40 and older reported poor sleep quality in 11.2% of participants as incident (new) cases and in 24.9% as persistent (ongoing) cases [11]. Sleep disorders such as insomnia and obstructive sleep apnoea remain underdiagnosed and undertreated [12]. These diagnostic and treatment gaps underscore the need for scalable and affordable technologies that enable the monitoring of sleep and behavioural interventions [13].

Sleep monitoring refers to the scientific analysis of sleep by measuring and analysing biological and behaviour-based cues [14]. This monitoring comprises objective approaches such as polysomnography (PSG), actigraphy, wearable devices, and subjective approaches such as sleep diaries and questionnaires. Sleep tracking is applied in clinical and research areas to determine sleep quality, to establish sleep structure, and to identify sleep disturbances.

Data-driven technologies, particularly Internet of Things (IoT) sensor systems, are increasingly used in health monitoring, including sleep monitoring. These systems capture real-time biometric data, facilitating personalised healthcare, disease prevention, and the early diagnosis of sleep and circadian rhythm disorders. Advances in accelerometers, photoplethysmography (PPG), and temperature sensors have enabled consumer devices to provide continuous and reliable monitoring of sleep disturbances and patterns [15].

The capabilities of sleep monitoring are further amplified with Artificial Intelligence (AI) [16]. AI refers to computer systems and procedures that are designed to perform tasks often requiring human intelligence, such as perception,

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

reasoning, learning, problem-solving, and decision-making [17]. In literature and academia, AI has sometimes been portrayed as the science and engineering of algorithms and programs that are used to create computers to simulate or mimic cognitive processes, with the goal of improving or replacing these cognitive processes.

AI enables predictive analytics, pattern recognition, and adaptive feedback in health technologies [16]. These technologies are particularly relevant in South Africa, where unequal healthcare access creates opportunities for digital tools to address gaps in clinical support. As one of these technologies, smartwatches, in particular, have globally become prime consumer health management tools [18]. Consumer-grade smartwatches are mass-market devices designed primarily for lifestyle monitoring (e.g., steps, heart rhythm, and sleep), compromising clinical accuracy for low cost and ease of use of the smartwatch [19]. By contrast, medical-grade or research-grade devices are thoroughly tested and approved by regulatory authorities, designed for clinical decision-making, and typically more accurate and reliable.

Global adoption of wearable devices has grown rapidly, and health and fitness applications have been the prime drivers of this expansion [20]. South Africa follows the same trend, with the adoption of wearables on the rise (predicted to be growing at a compound annual growth rate of 13.5% from 2025 to 2030) among tech-savvy users [21].

Recent studies on smartwatches highlight their expanding potential for preventing lifestyle-related diseases and aiding in providing access to health and sleep monitoring protocols [22]. Gaps exist in literature regarding the analysis of how these smartwatches implement different AI capabilities, using the required sensors, to analyse and improve sleep patterns.

The purpose of this study was to analyse the capabilities of the most popular consumer-grade smartwatches on the South African market to intelligently monitor sleep. In the rest of this paper, the term smartwatches will refer to consumer-grade smartwatches. The research question guiding this study is:

"How do popular smartwatch brands compare regarding AI-driven sleep monitoring capabilities?"

In answering this question, the study aims to enhance the evidence base for consumer health technologies and to understand their potential for adding to public health interventions to improve sleep health and reduce the NCD burden in South Africa.

A qualitative methodology with four stages of data collection and analysis was used. The smartwatch brands that were selected for analysis were the top five brands in South Africa in terms of popularity according to [23].

The structure of the paper is as follows: Section II explains the methodology, and Section III reviews the non-systematic literature. Section IV provides findings of the document analysis of white papers and validation studies of the different brands of smartwatches. Section V discusses validation studies of the different brands of smartwatches, and Section VI provides the results of the comparative study. Section VII discusses the smartwatch brands, AI features, sensors required, accuracy, and consumer trade-offs in using smartwatches for sleep tracking. Finally, Section VIII concludes the paper with recommendations for future research.

II. METHODOLOGY

The study followed a qualitative approach to answer the research question. The research had four stages: a non-systematic literature review, a document analysis, an analysis of existing validation studies, and a comparative analysis. Each stage focused on a different part of the research question and built upon the foregoing stage to create an integrated analysis.

In the first stage, the non-systematic literature review was carried out using the seven-step approach developed by [24]. The approach facilitated the integration of heterogeneous sources and the most recent literature. The literature was scrutinised under four areas: the accuracy of consumer-grade smartwatches versus PSG, sleep measurement methods, consumer-grade wearables for sleep, and the technological foundations for monitoring sleep through AI-driven capabilities and sensor integration. Through this literature review, a conceptual framework of AI-driven capabilities was constructed. This framework guided the later phases of the project. A conceptual framework is an analytical structure that defines the key concepts, variables, and presumed relationships in a study [25]. It provides the theoretical basis for the research and guides the formulation of objectives, methodology, and analysis. The conceptual framework was then used in later stages to analyse the AI-driven sleep monitoring capabilities of the five smartwatch brands. These brands were the five most popular South African brands for consumer-grade smartwatches: Apple, Samsung, Huawei, Fitbit, and Garmin [23]. Only the latest-generation models of smartwatches were compared to evaluate the differences in hardware and software.

The second stage involved the document analysis, using the qualitative method presented by [26]. Manufacturer specification sheets, peer-reviewed journal articles verifying the accuracy of smartwatches against PSG, and South African market reports were systematically searched for. The analysis was conducted using the two primary dimensions of:

- AI-driven capabilities: the algorithmic capability of the smartwatches to interpret sleep data; and
- Sensor requirements: the range and type of biometric sensors integrated into the smartwatch to collect physiological and behavioural parameters data.

These dimensions formed the high-level concepts in the conceptual framework that permitted smartwatches to be analysed based on technical ability and on potential contribution to scalable data-driven sleep health interventions in South Africa.

The third stage was used to examine existing validation studies from literature. Validation studies establish the reliability of consumer-grade smartwatches by cross-matching their sleep recordings with clinical gold standards, mainly PSG (and occasionally actigraphy) [27]. These studies are integral because the AI algorithms and sensors used by smartwatch manufacturers are not regulated as stringently as medical devices. Since no independent analysis is available, it is unknown whether consumer-grade smartwatches provide data credible enough for health monitoring or scientific research. Validation is paramount for

sleep science because the distinction of transitions from sleep into wakefulness, and the distinction of sleep and wakeful states, rely on the proper quantification of the physiological signals, such as brain wave activity, muscular tension, and eye movements. However, smartwatches measure only indirect proxies such as accelerometry and heart rate variability (HRV). These indirect proxies tend to lead to the smartwatches showing high sensitivity for sleep identification, but reduced specificity for the discrimination of wake states. The smartwatches clinically stage sleep with poor accuracy. Peer-reviewed studies that compared smartwatch sleep recordings to PSG (or actigraphy) were examined in this stage to determine the accuracy of the smartwatches. Sensitivity, specificity and agreement with PSG for each of the five smartwatch brands were extracted to assess how accurately the smartwatches detected sleep, wake states and sleep stages.

In the fourth stage, a comparative analysis was employed, drawing on the work of [28], and brought together the outcomes of the first three stages. The different smartwatch brands AI-driven capabilities and required sensor technologies were compared. Validation studies from peer-reviewed literature for each smartwatch brand from the previous stage were compared regarding accuracy of the smartwatches, reliability, and contextual limitations. Similarities and differences were determined across the smartwatches.

The primary data for this study was sourced from international technical white papers, peer-reviewed literature, and manufacturer websites. Since there were no locally derived South African validation or manufacturer-specific studies, the results are based on data from global studies and manufacturer-reported specifications rather than local studies.

III. Non-Systematic Literature Review

As sleep increasingly becomes a public health issue, PSG continues to represent the gold standard for measuring sleep, obtaining comprehensive data on sleep stages and physiology [29]. PSG is, however, expensive, resource-demanding, and available only in laboratory or clinic use, rendering it impractical for extended use in natural settings [30]. PSG's high cost-of-use means consumer wearables, including smartwatches and rings, can provide sleep-related data and sleep monitoring capabilities beyond clinical contexts. The key benefits of these wearables are low price, convenience, and familiarity with daily routines.

The scalability of consumer wearables, specifically smartwatches, offers an inherent possibility for sleep monitoring at a population scale, especially for NCDs related to improper sleep. However, studies of these wearables in African settings are largely deficient. Validation studies of these wearables have mostly been conducted in European, North American, and Asian populations, with African studies being scarce [30]. Closing the gap is vital for South Africa, where stress in urban regions, income inequity, and NCDs are increasing. Adoption of wearables, specifically smartwatches, increases among youth and urbanised populations, but low- and moderate-income communities are deterred from adopting wearables because of their high costs [22]. Moreover, data availability is often limited to proprietary platforms, creating ecosystem lock-in and stifling compatibility with public health infrastructures [30]. However, the capabilities provided by the smartwatches have the potential to outweigh these constraints.

An AI-driven capability uses computational techniques, including machine learning or adaptive algorithms, to make predictions or take actions that go beyond sensor measurements [31]. In health applications, AI-driven capabilities typically include disorder (e.g. sleep apnoea) risk screening, predictions, and personalisation [32]. Systems that generate fixed-rule or heuristic outputs that do not change with new data are described as non-AI capabilities.

AI-driven capabilities for sleep monitoring are split into seven different capabilities in four categories. In the first category are the foundational measurements: sleep staging and sleep quality scoring. Sleep staging divides sleep into three- or four-stage models such as light, deep, rapid eye movement (REM), and wake [33]. Sleep scoring or quality indexing produces a single score suggestive of quality, efficiency, and duration [34]. The second category contains the health risk and safety monitoring capabilities, namely disorder risk screening and anomaly detection. Sleep disorder risk screening is where algorithms make inferences such as apnoea or irregular breathing [35]. Anomaly detection covers alerting over unusual breathing, sleep fragmentation, and other unusual sleep pattern anomalies [36]. Personal profiling and classification is the third capability category, which includes chronotype classification and recovery assessment. Chronotype or sleep profile classification is an AI capability where subjects are organised into circadian types or sleep behaviour clusters [37]. Recovery assessment is an automated measure of an individual's physiological and cognitive restoration during and after sleep [38]. The last AI-driven capability is actionable insights consisting of customised coaching/feedback. Customised coaching or feedback provides behaviour-specific suggestions based on behavioural and sleep data [39].

These seven AI-driven capabilities are dependent on having various sensors embedded in the watch. Accelerometers are an important sensor technology, tracking body movement and activity patterns as the basis for actigraphy [40]. Cardiac-related sensor technologies include PPG, HR, and HRV sensors measuring cardiovascular dynamics [27]. SpO_2 sensors measure blood oxygen saturation, a metric essential in tracking apnoea risk [35]. Circadian rhythm and thermoregulation are determined through skin temperature sensors, with ECG providing superior cardiac monitoring [41], [42]. Microphones help identify snoring and environmental noise monitoring, while electrodermal activity (EDA) sensors measure stress responses through sympathetic nervous system activity [43]. Cardiopulmonary coupling (CPC) integrates heart and breathing data to evaluate autonomic regulation in sleep [44]. CPC is classified as a sensor due to it being a signal processing method instead of a higher-level AI-driven output. Figure I outlines the conceptual framework for categorising AI-based sleep tracking capabilities, upon which subsequent analysis was performed.

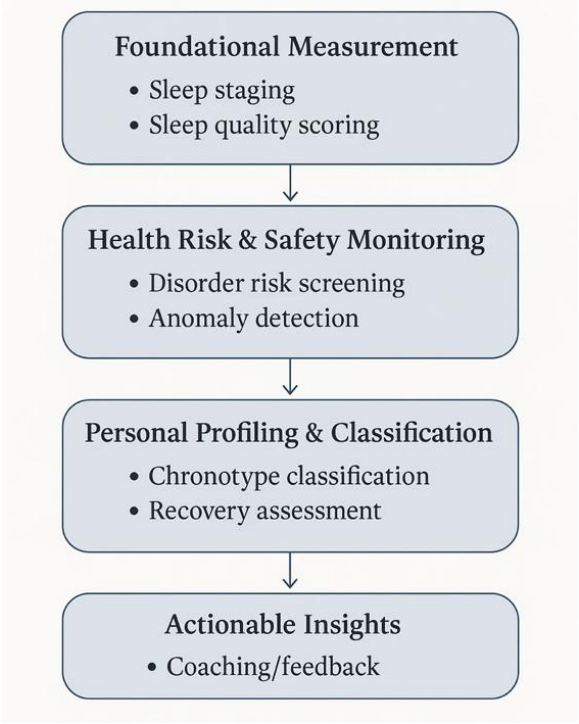


Figure I: Conceptual Framework for AI-driven Sleep Capabilities

AI models can process the raw physiological data collected from these sensors [45]. The AI combines these signals into meaningful measures such as sleep stages, quality scores, and reported measures that can be integrated into user-centred insights about their sleep behaviour [38]. The power of those signals lies in extracting sleep analysis, evaluating sleep recovery outcomes, and monitoring health behaviours using continuous non-intrusive data.

The categories of AI capabilities establish a conceptual framework for comparing smartwatches' sleep monitoring capabilities. Studies worldwide demonstrate promise and shortcomings for such technologies [46].

In spite of the numerous benefits that smartwatches and their capabilities can bring to their owners, there are still some concerns related to the comparative accuracy of consumer wearables, specifically smartwatches, as compared with that of PSG [27]. Accuracy studies repeatedly demonstrate that smartwatches are sensitive in recording sleep but not very good at discriminating wakefulness. A multicentre evaluation of 11 consumer wearables reported poor performance in sleep staging, as reflected in low macro F1 scores [30]. In the same study, a few smartwatches, such as the Google Pixel Watch and Fitbit Sense, had better performance in recording deep sleep. In contrast, others had systematic errors in estimating sleep efficiency or latency.

Device-specific evidence supports such weaknesses. Fitbit Charge 2 and Alta HR overestimated Total Sleep Time (TST) and underestimated wake time for obstructive sleep apnoea patients, decreasing utility in a clinical environment [47]. The Fitbit Charge 4 underestimated slow-wave sleep for subjects with insomnia and overestimated light sleep [48]. Such weaknesses represent the technical gap between wearables and PSG. Accelerometers dominate smartwatches, aided partly by multisensory inputs such as Heart Rate (HR) and HRV. In contrast, PSG measures neurophysiological activity directly, e.g. using electroencephalography (EEG) [49]. Although PPG and HRV added staging beyond this, misclassification from light to slow-wave sleep persists [50]. Broader issues in technical bias, inter-subject variability, BMI, and prevalence of sleep disorder impact accuracy [30].

IV. Findings: Document Analysis

The document analysis used the conceptual framework (Figure I) to identify which of the seven AI capability categories were implemented by each smartwatch brand, how these capabilities were described, and which sensor technologies supported these capabilities.

Apple

Apple smartwatches provide sleep staging and anomaly detection capabilities. However, it does not provide sleep quality scoring, disorder risk screening, chronotype classification, recovery assessment, or sleep coaching. Apple employs sleep staging algorithms based on 30-second epochs for the discrimination of wake, REM, light (core), and deep sleep [51], [52]. These algorithms are trained on large databases to enable clinical generalisation across populations. Output is presented within the Apple Health application, providing longer-term averages such as resting heart and respiratory rates. The Apple Health application provides risk screening through its sleep apnoea notification feature [53]. This feature examines breathing patterns using on-board sensors and notifies users of irregular breathing events during sleep to provide early warning signs of possible sleep-disordered breathing.

Apple smartwatches use three sensors to evaluate cardiovascular performance: a three-axis accelerometer, a gyroscope, and a PPG sensor to evaluate cardiovascular performance [54], [55]. Recent models such as the Series 9 and Ultra include SpO_2, skin temperature, and circadian rhythm monitoring [56], [57].

Samsung

Samsung provides sleep staging, quality scoring, disorder risk screening, coaching/providing feedback, chronotype classification, and anomaly, but lacks a systematic method for recovery assessment. The Galaxy Watch smartwatch line employs AI models for staging and derives an aggregate overall Sleep Score from variables such as SpO_2 and sleep duration [33], [35]. The "Sleep Coaching" program characterises chronotypes and offers behaviourally grounded recommendations. Samsung provides apnoea risk screening from pulse waveform and SpO_2 data [35].

Galaxy Watch models also include accelerometers, gyroscopes, PPG, SpO_2, ECG (BioActive Sensor), and skin temperature sensors [30], [42], [58], [59]. Snore detection is accomplished using a paired smartphone microphone [60].

Huawei

Huawei smartwatches employ six of the seven capabilities which are: sleep staging, quality scoring, disorder risk screening, anomaly detection, coaching/providing feedback, and recovery assessment, but it does not employ chronotype classification. Huawei's TruSleep algorithm, developed together with the Harvard Medical School, employs a combination of respiration coherence, movement, and heartbeat data to distinguish light and deep sleep, provide a sleep quality score, and detect up to six disorders such as disordered breathing and insomnia [61]. The algorithm does

not generate descriptive reports, but TruSleep offers behaviour-oriented recommendations.

Huawei smartwatches include accelerometers, PPG, SpO_2, and ECG sensors [61]. Additionally, CPC, also employed, merges PPG and HR to compute an autonomic nervous activity estimate [44]. CPC supports detecting anomalies and examining recovery, and offers an affordable alternative to costly sensors.

Fitbit

Fitbit smartwatches provide sleep staging, quality scoring, anomaly detection, coaching/feedback, chronotype classification, and recovery assessment, but does not employ disorder risk screening capabilities. Fitbit algorithms differentiate REM, core, and deep sleep and provide daily sleep scores [62]. Higher-end models complete the package with monthly Sleep Profiles, grouping consumers into their chronotypes, whilst also having snore and noise detection capabilities [63], [64]. Fitbit values accessibility by returning interpretable results and enabling longer-term viewing of trends [49], [65].

Fitbit smartwatch models house accelerometers, PPG, SpO_2, EDA and skin temperature sensors [27], [43]. Some Fitbit models also include the integration of a microphone for the sole purpose of snore analysis [66]. The sensors underlie the staging, scoring, anomaly detection, coaching, chronotype, and recovery capabilities.

Garmin

Garmin smartwatches employ sleep staging, quality scoring, and recovery assessment, but provides no disorder risk screening, anomaly detection, coaching/feedback, or chronotype classification. As an alternative to the deep learning algorithms employed by other brands, Garmin relies on heuristic, rule-based staging developed by Firstbeat Analytics [34], [67]. This rule-based staging is still classified as an AI capability due to it being able to adapt to new data. Outputs from this rule-based staging include sleep stage allocation and wellness variables such as stress, recovery, and the "Body Battery" score reflective of the recovery evaluation [68]. The sleep features employed by Garmin are for performance support and training optimisation, instead of clinical monitoring.

Garmin smartwatches have accelerometers, gyroscopes, and SpO_2 monitoring [69]. The smartwatches also contain HR monitoring, which makes use of their Elevate PPG sensors. Table I summarises the findings of the comparative analysis of the five smartwatch brands with an overview of their AI-based capabilities and coverage of required sensors.

Table I: Smartwatch Sleep Monitoring Comparison Matrix

Brand	AI Capabilities	Sensors
Apple	Sleep staging, Anomaly detection	Accelerometer, gyroscope, PPG, SpO_2, temperature, ECG
Samsung	Sleep staging, Sleep quality scoring, Disorder risk screening, Anomaly detection, Coaching/feedback,	Accelerometer, gyroscope, PPG, SpO_2, temperature, ECG, microphone
Huawei	Sleep staging, Disorder risk screening, Anomaly detection, Recovery assessment	Accelerometer, PPG, SpO_2, ECG, CPC
Fitbit	Sleep staging, Sleep quality scoring, Anomaly detection, Chronotype classification, Recovery assessment, Coaching/feedback	Accelerometer, PPG, SpO_2, temperature, microphone, EDA
Garmin	Sleep staging, Sleep quality scoring, Recovery assessment	Accelerometer, gyroscope, PPG, SpO_2

V. Findings: Validation Analysis

The findings from the analysis of validation studies reviewed the accuracy of each brand's smartwatches to detect sleep and wake states, the correspondence of the staging outputs with PSG, and areas of limitations across specificity, duration estimates, or population validation studies.

Apple

Validation studies have shown that the Apple Watch line performs reliably in the discrimination of sleep from wake states, with a sensitivity of 97% to 98% [70], [71]. However, specificity in recognition of sleep-time wakefulness is, nonetheless, poor; a feature of the majority of accelerometer-based monitors. Correspondence with PSG for sleep-stage determination is moderate, with a sensitivity of 50% to 86% (Robbins et al., 2024). REM and light sleep are routinely misassigned, reflecting the absence of direct measures of brain function [71].

Samsung

Samsung sensors have a sleep detection sensitivity of approximately 95% but a lower specificity (52%) for wake detection [58]. Sleep stage classification accuracy is approximately 65%, with difficulties differentiating between REM and deep sleep. Algorithm upgrades, like neural networks, have increased validation accuracy of sleep stages to 71.6% [33]. The advances in accuracy are improving, but there is still a gap between everyday use and clinical precision.

Huawei

Huawei smartwatch validation data is limited. Independent verification of the Huawei Watch GT2 released reported underestimating TST compared with actigraphy and a lack of correspondence with PSG, in the form of a Pearson correlation of 0.47 and an intraclass correlation of 0.25 [72]. The outcomes of this study reflect poor stage precision and duration estimation. Besides Chinese reports, only a few others have been conducted regarding Huawei smartwatches, and independent evaluation from more diverse populations is required [73].

Fitbit

Meta-analyses of Fitbit smartwatches showed a high sensitivity for sleep detection of 95–96% and a wake detection specificity of 58–69% [64]. PSG correspondence for stage classification is moderate, but Fitbit has a relatively stronger REM sleep estimation accuracy of 62–89% compared to its estimation of light sleep of 55–81% and slow-wave sleep of 36–89% [39], [74]. Older models overestimate TST and underestimate wake after sleep onset (WASO) considerably, particularly in insomnia patients. Current models with advanced staging algorithms correspond more with PSG, including a trivially small bias of less than 12 minutes for TST and WASO [49].

Garmin

Validation of Garmin smartwatches has continuously demonstrated excellent sensitivity (~98%) for the identification of sleep but very poor specificity for the identification of waking (~30%) [75]. The identification of sleep stages remains moderate, like other brands. Overall sleep duration is also overestimated, with mean biases of 27.8 minutes and up to 46.9 minutes [48], [75]. The findings again affirm that Garmin smartwatches are good for overall sleep–wake classification but not precise for detailed staging or duration estimation.

VI. Findings: Comparative Analysis

The comparative analysis compared the five brands, in terms of which AI capabilities were provided by each brand and the related integrated sensors that enable these capabilities. The comparison also included an analysis of how they were related to clinical or consumer use, and how the smartwatches position within different pricing tiers reflected accessibility, performance, or premium status.

The findings of the comparative analysis highlighted different strategies between manufacturers. Apple and Samsung have the broadest sensor coverage and clinically relevant staging, while Fitbit leans more toward usability and long-term behaviour change at a much lower cost. Huawei smartwatches are designed with consideration for affordability and distribution within China, but no objective validation of its smartwatches in other countries could be found. Garmin is different from other manufacturers due to its focus on athlete recovery over creating smartwatches that align with clinical applications.

While Apple and Samsung have the most clinically relevant smartwatches, they are costly. Fitbit compromises lower accuracy [Section V (D)] for accessibility, which makes Fitbit smartwatches appealing monitoring system for general population health that clinicians and researchers can use. Huawei is limited in its breadth of potential generalisability and validation, but has provided some low-cost, direct entry options into use for consumers. Garmin stands apart from other manufacturers as it isolates its products to support high-performance athletes and provide performance analytics for these athletes who may need performance-oriented metrics. However, it does not appear to have any responsive sleep disorder assessment or the ability to monitor clinical changes.

Price is driven by positional strategy. Apple and Samsung are in the premium positioning tier with the best sensor coverage and best-designed AI support. Fitbit is priced low-to mid-tier, focusing on accessibility and behaviour change data with an improved usability strategy. Huawei is mostly mid-tier, with some products straddling into the premium tier. Garmin is aligned closely with the high-performance athlete position, but is limited only to the endurance athletes, not with any clinical inclusion.

VII. Discussion

This study evaluated the AI-driven sleep tracking capabilities of the leading smartwatch brands in South Africa using the conceptual framework (Figure I). The framework organises seven AI capability categories: sleep staging, sleep quality scoring, disorder risk screening, anomaly detection, coaching and feedback, chronotype classification, and recovery assessment. These capabilities are within four higher-level categories: Foundational Measurement; Health Risk & Safety Monitoring; Personal Profiling and Classification; and Actionable Insights. Framing the outcomes in these categories enables one to distinguish between basic measurement functions common across smartwatch brands, advanced risk and anomaly tracking, personalised profiling, and capabilities capable of generating actionable feedback.

Three key findings emerged. Firstly, brands universally adopted the Foundational Measurement AI capabilities. Most brands universally implemented sleep stage and quality scoring (Apple does not implement sleep quality scoring). These capabilities formed the foundation of sleep tracking functionality. More advanced capabilities, such as disorder risk screening, chronotype classification and recovery monitoring, were selectively present and varied accordingly. Samsung and Huawei emphasised disorder risk screening, Fitbit and Samsung offered chronotype classification, and Garmin, Fitbit, and Huawei offered recovery monitoring. Secondly, sensor integration was a differentiating characteristic: Apple and Samsung implemented the most sensors, namely accelerometers, PPG, SpO_2, temperature and ECG; and hence possessed more exhaustive coverage of AI-driven capabilities. Fitbit and Huawei offered standard sets of sensors with unique add-ins such as CPC or EDA. At the same time, Garmin emphasised training and performance, thus fewer health-specific sensors. Thirdly, a trade-off was noted in terms of accuracy and applicability. Validation studies consistently revealed excellent sleep–wake detection sensitivity but poor wake state specificity and fair sleep stage accuracy. These findings testify to smartwatches as supplements for long-term tracking and behaviour change, not to clinically displace PSG.

The implications when contextualised to South Africa highlight three main trade-offs. The first is cost and capability: high-end smartwatches like Apple and Samsung have broader AI capabilities. However, they are beyond the affordability of large portions of the population. In contrast, mid-range smartwatches, i.e., Fitbit and Huawei, are satisfactory for individual tracking at a lower cost and thus are more relevant to public health contexts. The second is ecosystems and interoperability: closed platforms restrict interconnectivity with public health services and limit broader usefulness for data acquired through smartwatches. The third is orientation: Garmin's performance-related outputs reflect how design decisions are appropriate for

certain portions of users and less so for broader health requirements.

The conceptual framework developed in this study provides an understanding of why sleep monitoring research with smartwatches is appropriate despite limitations to precision. Rather than excluding smartwatches as imprecise tools, the framework places their strength in scalability, cost-effectiveness, and their capacity to support long-term self-monitoring. This framework shows how large-scale monitoring of sleep behaviour can support early risk detection in low-resource settings and thus contribute to the prevention of NCDs. There is, however, a known need to individually validate smartwatches within the context of South Africa.

VIII. CONCLUSION

In South Africa, while the incidence of NCDs and healthcare access inequities increases the demand for functional monitoring tools, several trade-offs are apparent. Premium smartwatches have more complete and accurate capabilities but are not readily available to many of the population. However, a few consumers' financial means, and mid-level smartwatches strike a more grounded balance of capabilities and price and are therefore desirable for more general population health applications. The orientation of a smartwatch (toward lifestyle, health, or performance) shapes its suitability for distinct user segments.

Future research should favour local validation of smartwatch functionality with South African cohorts and high-risk groups for whom global evidence might have less transferable relevance. AI algorithms also require further optimisation to boost staging accuracy and improve disorder risk screening coverage at the population scale. Policy frameworks must favour interoperability, affordability, and open access to data to support incorporation with public health system infrastructure. More importantly, consumer-grade smartwatches are best conceptualised as scalable tools supporting behaviour change, long-term sleep-related risk, and self-tracking at the population scale. By adopting the conceptual framework, researchers and policymakers can assess how smartwatches contribute to improving sleep health and, in turn, help reduce South Africa's NCD burden. The results presented in this paper can be used by smartwatch manufacturers to refine sensor integration and AI-driven features. Researchers can use the conceptual framework for related investigations into the potential of AI for validating sleep assessment models and expanding wearable health applications. In answering the research question, sleep monitoring can be enhanced to support well-being, contribute to NCD prevention, and align with South Africa's public health objectives.

ACKNOWLEDGMENT

We would like to thank Telkom and the Telkom Centre of Excellence for funding and making this research possible

REFERENCES

[1] [1] Addo, J., … Maher-Edwards, G. (2024). The Africa non-communicable diseases (NCD) Open Lab… Journal of Global Health, 14. https://doi.org/10.7189/JOGH.14.04065

[2] [2] Pilusa, T. D., … Maimela, E. (2025). Exploring patients' understanding of behavioral risk factors… BMC Public Health, 25(1). https://doi.org/10.1186/s12889-025-22514-9

[3] [3] Wand, H., Naidoo, S., Govender, V., & Moodley, J. (2025). Quantifying Hypertension and Cardiovascular Conditions in South Africa. Journal of Prevention. https://doi.org/10.1007/s10935-025-00833-2

[4] [4] Govender, I., & Sunnasy, A. (2025). The growing problem of obesity in South Africa. South African Family Practice, 67. https://doi.org/10.4102/SAFP.V67I1.6001

[5] [5] Cho, Y. (2020). Early development of bidirectional associations between sleep disturbance and diabetes. Diabetes and Metabolism Journal, 44(5), 668–670. https://doi.org/10.4093/dmj.2020.0198

[6] [6] Knaul, M., … Radbruch, L. (2025). The evolution of serious health-related suffering… Lancet Glob Health, 13. www.thelancet.com/lancetgh

[7] [7] Wu, L., … Wang, X. (2024). Sleep duration and mortality in patients with chronic noncommunicable disease… Environmental Health and Preventive Medicine, 29. https://doi.org/10.1265/ehpm.23-00249

[8] [8] Ramar, K., … Trotti, L. M. (2021). Sleep is essential to health: An AASM position statement. Journal of Clinical Sleep Medicine, 17(10), 2115–2119. https://doi.org/10.5664/jcsm.9476

[9] [9] Peltzer, K. (2012). Sociodemographic and health correlates of sleep problems… South African Journal of Psychiatry, 18(4), 150–156. https://doi.org/10.7196/SAJP.369

[10] [10] Stranges, S., … Kandala, N. B. (2012). Sleep problems: An emerging global epidemic? Sleep, 35(8), 1173–1181. https://doi.org/10.5665/sleep.2012

[11] [11] Peltzer, K. (2023). Incident and persistent poor sleep quality… Journal of Psychology in Africa, 33(6), 625–631. https://doi.org/10.1080/14330237.2023.2279377

[12] [12] Benjafield, A. V., … Malhotra, A. (2019). Estimation of the global prevalence and burden of obstructive sleep apnoea… The Lancet Respiratory Medicine, 7(8), 687–698. https://doi.org/10.1016/S2213-2600(19)30198-5

[13] [13] Robbins, R., … Jean-Louis, G. (2019). Sleep Tracking: a Systematic Review… Current Sleep Medicine Reports, 5(3), 156–163. https://doi.org/10.1007/s40675-019-00150-1

[14] [14] Yin, J., Xu, J., & Ren, T. L. (2023). Recent Progress in Long-Term Sleep Monitoring Technology. Biosensors, 13(3). https://doi.org/10.3390/bios13030395

[15] [15] Segun-Omosehin, O. A., … Adedinsewo, D. (2025). Evaluating the digital health technology landscape in sub-Saharan Africa… NPJ Cardiovascular Health, 2(1). https://doi.org/10.1038/s44325-025-00055-9

[16] [16] Osa-Sanchez, A., … Garcia-Zapirain, B. (2025). Wearable Sensors and Artificial Intelligence for Sleep Apnea Detection: A Systematic Review. Journal of Medical Systems, 49(1). https://doi.org/10.1007/s10916-025-02199-8

[17] [17] Xu, Y., … Zhang, J. (2021). Artificial intelligence: A powerful paradigm for scientific research. Innovation, 2(4). https://doi.org/10.1016/j.xinn.2021.100179

[18] [18] Köhler, C., … Salgado-Baez, E. (2024). The Value of Smartwatches in the Health Care Sector… Journal of Medical Internet Research, 26. https://doi.org/10.2196/58936

[19] [19] Mendt, S., … Maggioni, M. A. (2025). Laboratory comparison of consumer-grade and research-established wearables… Frontiers in Physiology, 16. https://doi.org/10.3389/fphys.2025.1491401

[20] [20] Huhn, S., … Barteit, S. (2022). The Impact of Wearable Technologies in Health Research: Scoping Review. JMIR mHealth and uHealth, 10(1). https://doi.org/10.2196/34384

[21] [21] Grand View Research. (n.d.). South Africa wearable technology market size & outlook. https://www.grandviewresearch.com/horizon/outlook/wearable-technology-market/south-africa

[22] [22] Muller, C. (2022). South African Millennial Consumers' Adoption Of Wearable Activity Trackers… AJPHES, 28(4), 275–298.

[23] [23] Statista. (2025, July). Most used eHealth tracker / smart watches by brand in South Africa… https://www.statista.com/forecasts/1371135/most-used-ehealth-tracker-smart-watches-by-brand-in-south-africa

[24] [24] Greetham, B. (2020). How to Write Your Literature Review. Bloomsbury Academic.

[25] [25] Luft, J. A., … Gardner, G. (2022). Literature Reviews, Theoretical Frameworks, and Conceptual Frameworks… CBE Life Sciences Education, 21(3), rm33. https://doi.org/10.1187/cbe.21-05-0134

[26] [26] Bowen, G. A. (2009). Document analysis as a qualitative research method. Qualitative Research Journal, 9(2), 27–40. https://doi.org/10.3316/QRJ0902027

[27] [27] Schyvens, A. M., … Verbraecken, J. (2025). A performance validation of six commercial wrist-worn wearable sleep-tracking devices… SLEEP Advances, 6(2). https://doi.org/10.1093/sleepadvances/zpaf021

[28] [28] Pickvance, C. G. (2001). Four varieties of comparative analysis. Journal of Housing and the Built Environment, 16.

[29] [29] Yang, J., … Gu, Z. (2025). Accuracy Bias and Factors Influencing Polysomnography and Consumer Sleep-Monitoring Device… Nature and Science of Sleep, 17, 1757–1768. https://doi.org/10.2147/NSS.S537489

[30] [30] Lee, T., … Kim, J. W. (2023). Accuracy of 11 Wearable… JMIR MHealth and UHealth, 11. https://doi.org/10.2196/50983

[31] [31] Perera, Y. S., … Abeykoon, C. (2023). The role of AI-driven soft sensors… Engineering Applications of Artificial Intelligence, 121. https://doi.org/10.1016/j.engappai.2023.105988

[32] [32] Chustecki, M. (2024). Benefits and Risks of AI in Health Care: Narrative Review. Interactive Journal of Medical Research, 13, e53616. https://doi.org/10.2196/53616

[33] [33] Silva, F. B., … Penatti, O. A. (2024). Sleep staging algorithm based on smartwatch sensors… Sleep Medicine, 119, 535–548. https://doi.org/10.1016/j.sleep.2024.05.033

[34] [34] Doherty, C., … Altini, M. (2025). Readiness, recovery, and strain… Translational Exercise Biomedicine. https://doi.org/10.1515/teb-2025-0001

[35] [35] Jung, H., … Joo, E. Y. (2022). Performance evaluation of a wrist-worn reflectance pulse oximeter during sleep. Sleep Health, 8(5), 420–428. https://doi.org/10.1016/j.sleh.2022.04.003

[36] [36] Lui, G. Y., … Breen, P. P. (2022). The Apple Watch for Monitoring Mental Health–Related Physiological Symptoms: Literature Review. JMIR Mental Health, 9(9). https://doi.org/10.2196/37354

[37] [37] Kaushik, P., … Smarr, B. (2024). Automated Chronotyping from a Daily Calendar using Machine Learning. https://api.semanticscholar.org/CorpusID:271064297

[38] [38] Jin, S., … Zhang, Q. (2024). SLEEPWISE… EMBS. https://doi.org/10.1109/EMBC53108.2024.10782937

[39] [39] Muthumanickam, T., … Athiraja, A. (2025). Optimizing Human Sleep Patterns Using AI-Driven Insights… ESIC 2025, 195–200. https://doi.org/10.1109/ESIC64052.2025.10962569

[40] [40] Stucky, B., … Landolt, H. P. (2021). Validation of fitbit charge 2 sleep… Journal of Medical Internet Research, 23(10). https://doi.org/10.2196/26476

[41] [41] Rentz, L. E., Ulman, H. K., & Galster, S. M. (2021). Deconstructing commercial wearable technology… Sensors, 21(15). https://doi.org/10.3390/s21155071

[42] [42] Park, S., Zheng, D., & Lee, U. (2025). A PPG Signal Dataset… Scientific Data, 12(1). https://doi.org/10.1038/s41597-025-05152-z

[43] [43] Mathunjwa, B. M., … Hsu, Y. L. (2025). A Comprehensive Review of Home Sleep Monitoring Technologies… Sensors, 25(6). https://doi.org/10.3390/s25061771

[44] [44] Lu, M., Penzel, T., & Thomas, R. J. (2022). Cardiopulmonary Coupling. In Advances in the Diagnosis and Treatment of Sleep Apnea… Springer. https://doi.org/10.1007/978-3-031-06413-5_11

[45] [45] Hibbing, P. R., & Khan, M. M. (2024). Raw Photoplethysmography as an Enhancement… JMIR mHealth and uHealth, 12. https://doi.org/10.2196/57158

[46] [46] Herberger, S., Aurnhammer, C., Bauerfeind, S., Bothe, T., Penzel, T., & Fietze, I. (2025). Performance of wearable finger ring trackers for diagnostic sleep measurement in the clinical context. Scientific Reports, 15(1). https://doi.org/10.1038/s41598-025-93774-z

[47] [47] Moreno-Pino, F., … Baca-García, E. (2019). Validation of fitbit charge 2 and fitbit alta hr… Journal of Clinical Sleep Medicine, 15(11), 1645–1653. https://doi.org/10.5664/jcsm.8032

[48] [48] Schyvens, A. M., … Verbraecken, J. (2024). Accuracy of Fitbit Charge 4, Garmin Vivosmart 4, and WHOOP Versus PSG: Systematic Review. JMIR mHealth and uHealth, 12(1). https://doi.org/10.2196/52192

[49] [49] Dong, X., … Li, Y. (2022). Validation of Fitbit Charge 4… PLoS ONE, 17(10). https://doi.org/10.1371/journal.pone.0275287

[50] [50] Liu, P. K., … Lee, P. L. (2023). Validation of PPG- and acceleration-based sleep staging… Journal of Clinical Sleep Medicine, 19(10), 1797–1810. https://doi.org/10.5664/jcsm.10690

[51] [51] Caputo, D., … Cheng, P. (2024). Measuring Sleep with the Apple Watch… SLEEP, 47(1). https://academic.oup.com/sleep/article/47/Supplement_1/A123/7654069

[52] [52] Walch, O., … Goldstein, C. (2019). Sleep stage prediction with raw acceleration and PPG… Sleep, 42(12). https://doi.org/10.1093/sleep/zsz18

[53] [53] Apple. (2024). Estimating Breathing Disturbances and Sleep Apnea Risk from Apple Watch (White Paper). https://www.apple.com/health/pdf/sleep-apnea/Sleep_Apnea_Notifications_on_Apple_Watch_September_2024.pdf

[54] [54] Jaworski, D. J., & Park, E. J. (2023). Apple Watch Sleep and Physiological Tracking… EMBS. https://doi.org/10.1109/EMBC40787.2023.10340725

[55] [55] O'Grady, B., … Doherty, C. (2024). The Validity of Apple Watch Series 9 and Ultra 2… Sensors, 24(19). https://doi.org/10.3390/s24196220

[56] [56] Lee, E. (2025). Wearable Technology in Circadian Rhythm Research… Chronobiology in Medicine, 7(1), 3–8. https://doi.org/10.33069/cim.2025.0011

[57] [57] Shapiro, I., … O'Reilly, M. (2023). Pulse oximetry values from 33,080 participants… NPJ Digital Medicine, 6(1). https://doi.org/10.1038/s41746-023-00851-6

[58] [58] Kim, D., Joo, E. Y., & Choi, S. J. (2023). Validation of the Samsung Smartwatch… Journal of Sleep Medicine, 20(1), 28–34. https://doi.org/10.13078/jsm.230004

[59] [59] Wagner, G. R., Ibarra, K., & Mathew, A. (2024). Thermal Testing and Analysis of On-the-Market Smartwatches. ITHERM. https://doi.org/10.1109/ITherm55375.2024.10709507

[60] [60] Samsung. (2025). Use Samsung Health sleep monitoring on your Galaxy Watch. https://www.samsung.com/us/support/answer/ANS10003657/?utm_source=chatgpt.com

[61] [61] Huawei. (2025). HUAWEI Band 8 Principles behind sleep tracking. https://consumer.huawei.com/en/support/content/en-us15963854/

[62] [62] Young-Lin, N., … Lafon, B. (2025). Insights into maternal sleep… EBioMedicine, 114. https://doi.org/10.1016/j.ebiom.2025.105640

[63] [63] Fitbit Community. (2025). The Science Behind Fitbit's Sleep Profile animals. https://community.fitbit.com/t5/The-Pulse-Fitbit-Community-Blog/The-Science-Behind-Fitbit-s-Sleep-Profile-animals/ba-p/5630482

[64] [64] Haghayegh, S., … Castriotta, R. J. (2019). Accuracy of wristband fitbit models in assessing sleep… Journal of Medical Internet Research, 21(11). https://doi.org/10.2196/16273

[65] [65] Nestor, B. A., … Kossowsky, J. (2025). Evaluating Fitbits for Assessment of Physical Activity and Sleep in Pediatric Pain… JMIR Formative Research, 9. https://doi.org/10.2196/59074

[66] [66] Fitbit Help Center. (2025). How do I track my sleep with my Fitbit device? https://support.google.com/fitbit/answer/14236407?hl=en&utm_source=chatgpt.com#zippy=%2Chow-to-turn-on-snore-and-noise-detection

[67] [67] Maker, R. (2020, June 30). Garmin Acquires Firstbeat Analytics… https://www.dcrainmaker.com/2020/06/acquires-firstbeat-analytics.html?utm_source=chatgpt.com

[68] [68] Budig, M., Stoohs, R., & Keiner, M. (2022). Validity of Two Consumer Multisport Activity Tracker… Sensors, 22(23). https://doi.org/10.3390/s22239540

[69] [69] Helmer, P., … Sammeth, M. (2024). Reliability of continuous vital sign monitoring… Digital Health, 10. https://doi.org/10.1177/20552076241254026

[70] [70] Apple. (2023). Estimating Sleep Stages from Apple Watch. https://www.apple.com/healthcare/docs/site/Estimating_Sleep_Stages_from_Apple_Watch_Sept_2023.pdf

[71] [71] Robbins, R., … Duffy, J. F. (2024). Accuracy of Three Commercial Wearable Devices for Sleep Tracking… Sensors, 24(20). https://doi.org/10.3390/s24206532

[72] [72] Mei, L., He, Z., & Hu, L. (2024). Accuracy of the Huawei GT2 Smartwatch… JMIR Formative Research, 8. https://doi.org/10.2196/59521

[73] [73] Guo, Y., Zhang, H., & Chen, Y. (2021). Population screening for atrial fibrillation in subjects with sleep apnea. European Heart Journal, 42(Suppl 1). https://doi.org/10.1093/eurheartj/ehab724.0454

[74] [74] de Zambotti, M., … Baker, F. C. (2018). A validation study of Fitbit Charge 2TM… Chronobiology International, 35(4), 465–476. https://doi.org/10.1080/07420528.2017.1413578

[75] [75] Mouritzen, N. J., … Kjær, T. W. (2020). Assessing the performance of a commercial multisensory sleep tracker. PLoS ONE, 15(12). https://doi.org/10.1371/journal.pone.0243214

MobileNetV3-Based Disease Detection from Poultry Faecal Images Using a Mobile-Integrated System

Tinashe Crispen Garidzira
Department of Computational Sciences
University of Fort Hare
Alice, South Africa
0009-0003-4850-1568

Khulumani Sibanda
Department of Applied Informatics and Mathematical Sciences
Walter Sisulu University
Mthatha, South Africa
ksibanda@wsu.ac.za

Nyashadzashe Tamuka
Department of Economics and Finance
University of the Free State
Bloemfontein 9300, South Africa
tamukane@ufs.ac.za

***Abstract*— This research describes the development and evaluation of a mobile-friendly system designed for the early detection of poultry diseases through the analysis of faecal images. Utilizing transfer learning with MobileNetV3, the suggested deep learning model achieves an accuracy rate of 96.07% in classifying prevalent poultry diseases, including Coccidiosis, Salmonella, Newcastle, as well as healthy states. The model is incorporated into a FastAPI backend, allowing for instantaneous inference via an intuitive mobile application called "Poultry Scope," which supports on-site disease diagnosis with an average inference duration of 100 milliseconds. This system overcomes current limitations in disease detection techniques by offering an effective, scalable, and accessible solution for poultry farmers, which encourages prompt interventions and enhances animal health management. Future developments will aim to diversify the dataset, incorporate object detection methods, and implement the system in areas with low connectivity to improve its resilience and usefulness across varied farming environments.**

Keywords*— *Poultry Disease Detection (PDD), Precision Agriculture, Deep Learning, Livestock Health Monitoring

I. INTRODUCTION

The fields of precision agriculture and livestock farming have evolved rapidly in recent years, driven by the need for improved animal health management, reduced economic losses, and enhanced food safety [1], [2], [3]. The poultry industry is a primary source of protein for humans; failing to meet protein demands can negatively impact the economy and food security in any country [4], [5]. It is therefore exigent to have systems and methods for the early detection of diseases in poultry to prevent huge losses of poultry birds. According to Reyad et al. [5], disease detection from chicken faecal images represents a significant step towards automation in the poultry industry. Artificial intelligence (AI) and computer vision have become crucial tools for revolutionising disease detection in poultry and improving the Agriculture Sector [6], [7]. Traditional methods rely on manual inspection, which is time-consuming, labour-intensive, and less accurate, especially for large-scale farmers [4], [5], [8]. According to Hassan et al. [8], early detection and identification of disease from images using Machine Learning (ML) has emerged as an important approach to poultry disease detection because of its reliability, scalability, and better accuracy [4].

Currently, if farmers suspect health issues with their chickens, they seek professional help, and getting results and advice often takes long, as samples have to be analysed in laboratories [4], [7], [9]. This process is inherently slow as it depends on the availability of experts and laboratory infrastructure. In poultry, diseases can be directly detected from the faecal samples; however, this process heavily depends on the experience of the poultry farmer [10]. Early detection of disease can result in early treatment and the prevention of the spread. The most common diseases in poultry are Coccidiosis, Salmonella, and Newcastle, which, if not detected early, can lead to mass death of chickens [11]. A quick and precise health assessment of poultry can assist producers in making time-sensitive decisions to reduce disease spread, improve animal health, and boost the economy [3]. AI and ML have shown great potential in early disease detection in poultry using image processing. According to Sichilima et al. [7], smart farming solutions, particularly data-driven solutions, have proven to be effective predictive analytical tools for large data volumes, and these solutions are found in different domains such as medicine, finance, and sports, and now increasingly in agriculture.

Recent advances in computer vision and deep learning have enabled automated image-based disease classification in poulltry. These techniques offer the potential for fast, low-cost and scalable screening, particularly when integrated into mobile devices already available to many farmers. However, to be practically useful in the field, diagnostic models must be both accurate and computationally efficient, with inference times suitable for real-time use on resource-constrained devices. This study addresses the problem of enabling rapid, on-demand classification of poultry faecal images into disease categories using a lightweight deep learning model that can be integrated into a mobile application for farm-level decision support. The aim of this study was to design, train, establish and deploy an efficient MobileNetV3 capable of classifying poultry faecal images into predefined disease classes and a healthy class, and to expose this capability via a production-oriented API suitable for mobile integration. The contributions of this paper are:

1. To fine-tune a MobileNetV3-based architecture on a labelled poultry faecal images dataset, achieving high classification performance while maintaining a compact model size suitable for deployment
2. To provide a complete deployment pipeline using FastAPI, enabling real-time inference with low latency that can be consumed by mobile or web clients
3. To evaluate the performance of the developed model and illustrate how farmers could use the system for decision support.

The rest of this paper is organized as follows: Section II reviews related works to this study, while Section III presents the methodology adopted in this study. Section IV reports and discusses the experimental results, and Section V concludes

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

the study by highlighting key findings and outlining directions for future research.

II. Related Works

Recent studies have explored image-based disease classification in poultry. Sichilima et al. [7] proposed a Convolutional Neural Network (CNN) based framework to classify poultry diseases by distinguishing healthy and unhealthy faecal images. The authors used faecal images to classify them as diseased or healthy. Their model obtained an accuracy of 84.99% on the training set and 90.05% on the testing set. However, their study have some limitations for example, it did not consider building user-friendly solutions such as web interfaces or mobile applications, which can enable real-time and on-the-field detection.

Degu et al. [9] proposed a two-stage pipeline for poultry disease detection (PDD) and recognition from faecal images. They used YOLOv3 to localize regions of interest (ROI) in field images of chicken faeces, followed by ResNet-50 using Transfer Learning (TL) to classify four health conditions (healthy, coccidiosis, salmonella, Newcastle). They trained the classification model on 10,500 annotated images with augmentation and oversampling to mitigate class imbalance. The system obtained 87.48% mAP for ROI detection and 98.7% classification accuracy on the segmented patches, demonstrating high accuracy but not without increased reference latency and complexity. While their approach proved effective, its reliance on multiple models could come with deployment complexity and increased maintenance demands.

Machuve et al. [11] compared four TL architectures, the VGG16, InceptionV3, MobileNetV2, and the Xception, using farm-sourced images. The authors showed that fine-tuning with batch normalization layers, frozen substantially improved performance, and they recommended MobileNetV2 for mobile deployment due to its favourable accuracy efficiency trade-off. However, the focus of their study was limited to cropped faecal images without an explicit detection step for mixed scenes, which is not a true reflection of the real world. Additionally, their study was geographically confined to northern Tanzania, which can limit its wide applicability.

Li et al. [3] developed a prototype that integrated a modified MobileNetV2 for onsite faecal classification and achieved an accuracy of over 90%. Their systems showed the feasibility of near-real-time decision support, although it may be problematic in low-connectivity environments, such as in rural areas, because of its dependence on server-based inference.

Despite great strides made by previous studies, there is still a gap in integrating MobileNetV3 through TL with a FastAPI server connected to a mobile application for real-time disease prediction from poultry faecal images. The study by Sichilima et al [7] showed significant findings in poultry disease diagnosis but did not integrate a user interface application, while Degu et al. [9] proposed a more complex two-stage approach that introduced latency during inference and increase maintenance challenges compared to our simple pipeline. Similarly, although Machuve et al. [11] employed TL to enhance classification performance, their study was geographically limited, focusing primarily on poultry farms in Tanzania, which constrains broader applicability. On the other hand, Li et al. [3] developed a The

Recent research documented in [12] focused on high-risk disease detection in poultry using TL. The authors trained six different TL architectures, namely DenseNet201, Resnet152V2, InceptionV3, InceptionResnetV2, MobileNetV2, and Xception, selected for their widespread use and high accuracy rates. Their study showed that MobileNetV2 achieved the highest accuracy rate of 97.1% and exhibited the fastest training times. A key strength of their study was its explicit focus on faecal imagery and a transparent comparison across CNN architectures. However, the study was constrained by a geographically limited dataset, raising concerns about its robustness to lighting variation, flooring, and camera variability in other farms.

Reyad et al. [5] used an ensemble DL approach to assess poultry health from faecal image analysis. They evaluated five different CNN architectures: EfficientNetB7, VGG19, MobileNetV3, Vision Transformer, and Swin Transformer, using the poultry disease dataset from Kaggle. Their findings demonstrated classification accuracies of 95% for EfficientNetB7, MobileNetV3 91%, VGG19 96%, and 99% for both Swin and Vision Transformers. The key strength of their study was the comparative analysis of different architectures, achieving a production-ready accuracy with a Transformer-based Model of 99%. Nonetheless, the study experienced limitations of computational complexity and possible overfitting, particularly regarding cross-farm generalization. Additionally, their study did not incorporate deployment strategies as well as integrate mobile applications to enhance usability.

Chidziwisano et al. [6] studied the generalizability of DL for poultry disease prediction using faecal data from Tanzania and Malawi. The authors developed models using common CNN architectures: MobileNet, DenseNet, and ResNet. They trained the models using data from Tanzania and subsequently evaluated them using unseen data from Malawi. Their findings indicated that MobileNet binary, MobileNet multiclass, DenseNet binary, and DenseNet multiclass classifiers achieved an accuracy of 98%, 91%, 98%, and 95% respectively. This study's strength lies in However, it predominantly focused on faecal data prediction without extensive discussion on mobile deployment or end-user interfaces. The strength of their study is in its cross-country analysis and comparative evaluation of various CNN models. However, their work emphasizes prediction from faecal data with limited detail on mobile deployment or end-user interfaces.

Anwarul et al. [13] proposed a "DeepFowl", a DL based method for disease detection from chicken excreta images. They modified DenseNet-169 with a bespoke oversampling strategy to address class imbalance. The results showed an accuracy of 98.4% and demonstrated that deeper backbones CNNs can excel in faecal-image classification, with clear methodological transparency and effective data balancing. However, the study's limitations include the potential computational load of DenseNet-based models and the lack of deployment for end-systems.

Hasan et al. [14] proposed an image-based detection and classification system for chicken droppings in open-house farms in Malaysia. They combined dropping detection with downstream health-condition classification. They used the YOLOv5n algorithm for detecting chicken droppings and distinguishing between healthy and sick, based on observable features such as the droppings' colour and shape. Their proposed pipeline demonstrates the value of scene-level detection before classification, reporting a good detection and classification accuracy rate of up to 94.9%. A notable strength in their study was the end-to-end framing suited to real barns. However, environmental variation such as lighting and floor substrates across farms may challenge generalization without additional domain-adaptation strategies.

Study by Akbudak in [15] proposed a mobile application aimed at assisting poultry farmers in early identification of pathogenic diseases. He trained DL models on faecal images to automate disease classification, replacing slow and costly laboratory-based diagnostic methods [4], [7], [9] The results demonstrated that the proposed mobile application could provide poultry producers with fast and reliable information about the health status of their flocks at their fingertips. This mobile integration helps in reducing delays that often worsen the mortality rates of poultry. The strength of this study lies in its practical orientation, directly addressing farmers' needs by embedding disease detection into a mobile platform for accessible and immediate decision-making. However, its limitation was the lack of extensive validation on larger, diverse datasets and the need for further optimization of model robustness to handle variable field conditions and faecal image quality.

The reviewed related works show the promising potential of DL and TL approaches for poultry disease detection from faecal images. Despite all the strengths of the works, there are prevailing limitations that include reliance on limited or geographically constrained datasets, high computational complexity, and a lack of deployment in real farm environments. Even though mobile-based solutions have been attempted for exanple Akbudak [15], they require stronger validation in diverse environments. Thus, there remains a clear need for mobile-integrated, lightweight, and generalizable systems that can ensure robust and real-time disease detection across varying farm conditions.

III. METHODOLOGY

This section outlines the detailed methodology used in creating a mobile application designed to identify diseases through the analysis of poultry faecal samples. The research followed the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework, which includes six essential stages: understanding the data, preparing the data, feature engineering, model development, evaluation, and deployment [16]. Using a publicly accessible dataset from Kaggle, created by [17] and named Poultry Diseases Detection, a complete disease classification system was developed utilizing DL, FastAPI, and React Native technologies.

The datase's colour distribution was described by computing the mean and standard deviation for each colour channel (Red, Green, Blue) across all images. These statistics are presented in Table I, which acts as normalization parameters to guarantee consistent input distributions throughout the training, validation, and testing stages.

TABLE I. CALCULATED MEANS AND STANDARD DEVIATIONS OF COLOUR CHANNELS.

	Red (R)	Green (G)	Blue (B)
Mean	0.5892	0.5628	0.5127
Standard Deviation	0.1514	0.1529	0.1645

Normalization was achieved by taking the mean away from each value and dividing by the standard deviation for every channel, which helped to align the distributions of input data and led to quicker convergence and enhanced stability of the model [18], [19].

The dataset was partitioned into three segments: 80% allocated for training and 20% for testing purposes. The training segment was additionally divided, with 20% reserved for validation to optimize hyperparameters and assess the model's generalization performance. To enhance the model's robustness and ability to generalize, a range of data augmentation methods, such as resizing, random horizontal flips, random cropping, random rotations, and normalization, were applied to the training data using torchvision transforms [20]. In contrast, the validation and testing datasets were only subjected to resizing, normalization, and conversion into tensors to maintain the integrity of the evaluation and prevent any data leakage [21], [22]. All images were normalized using the precomputed mean and standard deviation values listed in Table I during both the training and inference stages, which ensured consistent model training and efficient inference [23]. Fig. 1 depicts the distribution of samples across the different splits of the dataset.

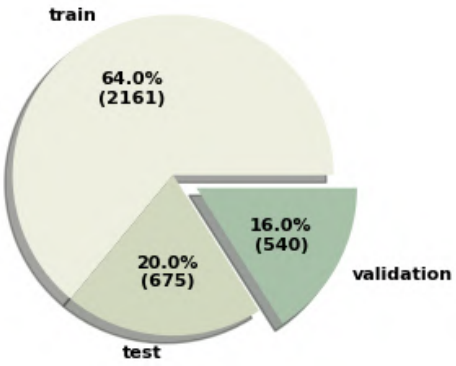


Fig. 1. Sample composition across sets.

Fig. 1 shows the composition of samples in each subset, with 64% for training, 20% for testing, and 16% for validation. Each subset was batched to ensure efficient data loading during training [24]. To address dataset imbalance particularly the smaller number of Newcastle disease images augmentation methods such as rotation, horizontal flipping, and random cropping were applied [20], [23], [25]. Moreover, class weighting was integrated into the loss function to minimize the misclassification of minority classes [24]. The DataLoader utility from torchvision was used with a batch size of 64, which improved computational efficiency [24]. Fig. 2 shows sample images from the dataset for both healthy and diseased faeces.



Fig. 2. Sample images in the dataset.

Images with red titles represent faeces from diseased poultry, while images with green titles represent faeces from healthy poultry, as illustrated in Fig. 2. Table II shows the model's parameters.

TABLE II. MOBILENETV3 PARAMETER COUNTS.

Parameters	Count
Total	4,207,156
Trainable	4,207,156
Non-Trainable	0

Table II shows the parameters of the MobileNetV3 model, which has more than 4.2 million trainable and total parameters, with no non-trainable parameters. TL with fine-tuning was used by modifying the output layer to predict five (5) classes instead of the default 1000 [26]. Unlike approaches where models are partially frozen, we did not freeze the layers, allowing the network to retrain and adapt its weights specifically to poultry faeces features [27]. The MobileNetV3-Large architecture was adopted for its optimal balance between latency and accuracy [27]. Input dimensions were 224x224x3, with training conducted using the Adam optimizer at an initial learning rate of 0.001 and a batch size of 64. All layers were unfrozen for fine-tuning, enabling the model to adapt to domain specific faecal image features.

The model was trained for 10 epochs, and performance was monitored using metrics such as training and validation accuracy, training and validation loss, and epoch-wise training time [25]. To preserve the best model weights, model checkpointing was used, where the model is saved as the best-performing model based on validation loss [28]. The best saved model was evaluated on the test dataset using accuracy, loss, confusion matrix, and classification reports, ensuring robust performance before deployment with the FastAPI server [25], [29]. Fig. 3 illustrates the end-to-end disease detection architecture

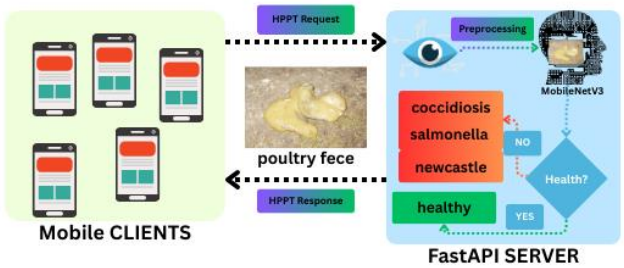


Fig. 3. End-to-End Disease Detection System.

Fig. 3 shows the end-to-end disease detection architecture, employing a basic client-to-server design in which the Poultry Scope native mobile application (iOS and Android) acts as the client. The client sends HTTP POST requests containing faecal images to a FastAPI backend and receives results via HTTP responses, as shown in Fig. 3. FastAPI is a modern high-performance Python web framework, enables asynchronous handling and low-latency model inference [29]. After receiving the image from the request body, the server then adds the image to the preprocessing pipeline, in which the image is resized to 224x224px, then converted to a tensor, and normalised based on precomputed means and standard deviations, as shown in Table I. The processing pipeline that is used during this inference process is the testing pipeline.

The pre-processed image is then passed to the best saved MobileNetV3 model chosen for its lightweight architecture optimized via platform-aware [26]. This model outputs five logits corresponding to 5 output classes, then they are transformed using the SoftMax activation function. The top-class label is the index of the 5-dimensional output with the highest probability. The predictions are then returned to the client as a JSON response. This simple pipeline demonstrated real-time classification capability with an average inference time of 100 milliseconds, showing effective deployment of DL models in mobile applications [30], [31].

IV. RESULTS AND DISCUSSION

This section outlines the results obtained throughout the creation, training, and assessment of the proposed model. The analysis starts with a summary of the training history, highlighting essential metrics such as the duration of each epoch, loss, and accuracy for both the training and validation datasets [20], [25]. Furthermore, the overall training time was documented to evaluate computational efficiency [32].

After visualizing the training process, we assessed the performance of the best model checkpoint using several metrics, specifically testing loss, accuracy, classification report, and confusion matrix, to thoroughly evaluate classification performance [25]. Sample predictions from the test dataset are included, along with a figure that highlights the images that were most commonly misclassified. In addition, screenshots of the prototype for the Poultry Scope mobile application are provided to showcase its practical deployment capabilities [15]. Table III presents a summary of the training metrics.

TABLE III. MOBILENETV3 TRAINING HISTORY.

Metric	Value
Total Epochs	10
Last save Epoch	8
Total Training Time	28 minutes 43.92 seconds

The cumulative training time for 10 epochs was around 28 minutes and 44 seconds. The model checkpoint for epoch 8 was deemed the best, as it showed the minimum validation loss, which aligns with standard practices for early stopping and checkpointing [28]. Fig. 4 illustrates the training durations for each epoch, offering a look into computational efficiency.

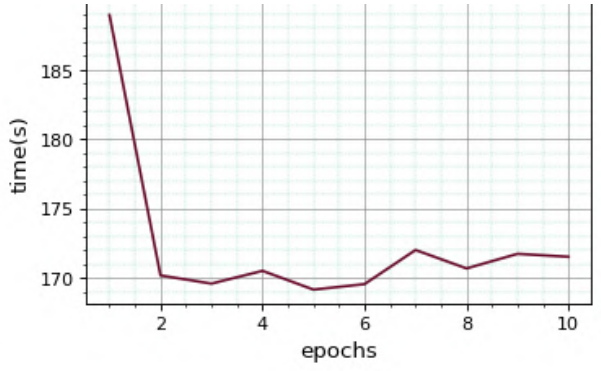


Fig. 4. MobileNetV3 epoch training time.

An analysis of Fig. 4 reveals that the first epoch took more than 185 seconds, most likely due to the delays in data loading. In the following epochs, a significant decrease in time was observed, stabilizing under 175 seconds, indicating that effective data caching in PyTorch led to enhanced data loading efficiency [24].

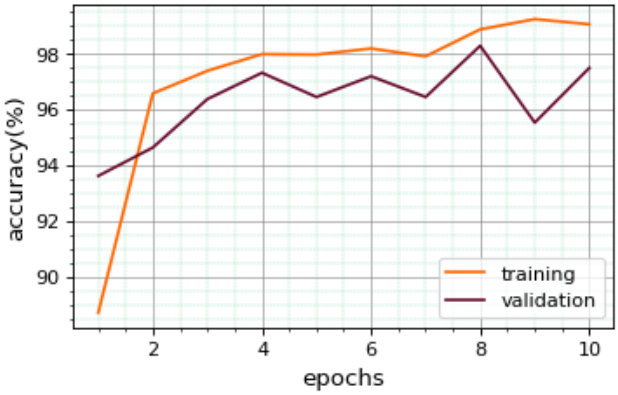


Fig. 5. MobileNetV3 training and validation accuracies.

Fig. 5 shows the development of training and validation accuracy throughout the epochs. Initially, training accuracy was below 90% in the first epoch, but it progressively improved, while validation accuracy exceeded 92% early in the training process, demonstrating the benefits of TL. After epoch 4, validation accuracy showed some variability, peaking at epoch 8.

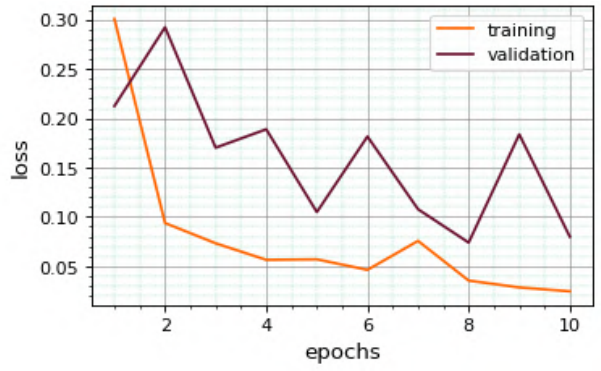


Fig. 6. MobileNetV3 training and validation losses.

The loss trajectories are illustrated in Fig. 6. The training loss steadily diminished from roughly 0.3 to under 0.05, demonstrating successful learning. The validation loss, which started at about 0.25, experienced variations with a low point at epoch 8, indicating that this particular checkpoint provides the most effective generalisation. The assessment of the optimal model produced the metrics outlined in Table IV, demonstrating excellent performance on unfamiliar data.

TABLE IV. BEST SAVED MODEL LOSS AND ACCURACY.

Metric	Value
Loss	0.16
Accuracy	96.07%

The best checkpointed model achieved an accuracy of 96.07% with a minimum loss of 0.16 as shown in Table IV. The model's low test loss and high accuracy highlight its reliability and suitability for use in practical applications [28]. A comparative analysis with Degu and Simegn [9], who used the same Kaggle dataset with a YOLOv3-ResNet pipeline, reveals that our MobileNetV3 model achieves a comparable 96.07% accuracy versus their 98.7%. Although their approach benefits from a two-stage detection mechanism, our single-pipeline system offers faster inference and lower deployment complexity, making it more suitable for mobile integration. The confusion matrix depicted in Fig. 7 shows the classification results for each class.

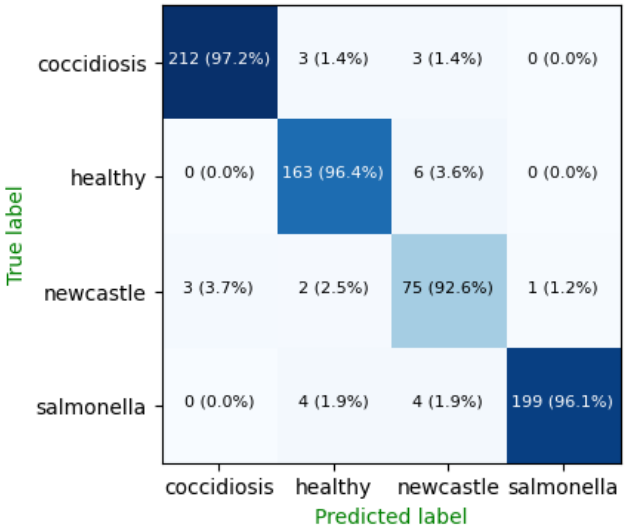


Fig. 7. Best saved model Confusion Matrix.

The model achieved a classification accuracy of over 92% for all categories, with only slight misclassifications mainly happening between visually similar groups, such as healthy vs. coccidiosis and salmonella vs. other groups. These mix-ups highlight the natural difficulties in distinguishing between similar pathological characteristics [6]. Fig. 8 summarizes class wise precision, recall, and F1-scores.

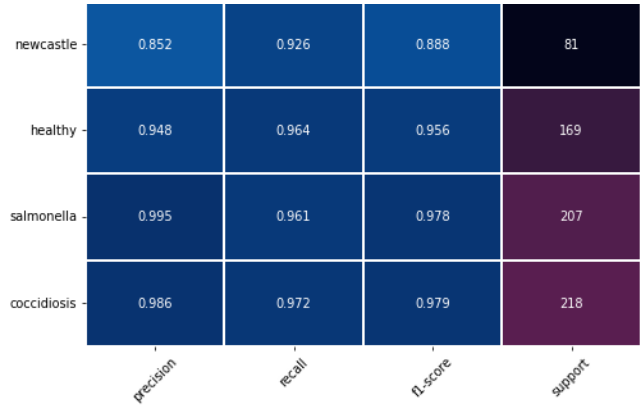


Fig. 8. Best saved model classification report.

In general, the model reached an accuracy rate of 96%, demonstrating especially effective performance in recognising coccidiosis, salmonella, and healthy samples. The somewhat lower performance on the Newcastle class highlights the challenge of identifying subtle visual distinctions, which is a frequent obstacle in image-based disease classification [33]. All classes achieved F1-scores above 0.90, with the Newcastle class slightly lower at 0.88, highlighting the influence of data imbalance. These metrics confirm that while the model performs robustly overall, further dataset expansion could enhance minority-class recognition



Fig. 9. Sample prediction of the best saved model.

Fig. 9 shows the sample predictions, with correctly identified images highlighted in green and misclassified ones in red. The model accurately classified all samples, though a few errors were noted in the more difficult cases. Fig. 10 shows the most mis-predicted images in the test dataset.



Fig. 10. The best saved model most incorrect predictions.

Fig. 10 illustrates the images that exhibit the greatest misclassification rates, particularly the confusion between healthy samples and those affected by coccidiosis, which underscores the necessity for a broader range of training data.

Ultimately, Fig. 11 displays screenshots of the prototype for the "Poultry Scope" mobile application, highlighting some its user interface and features. The application interacts with the FastAPI backend to provide real-time predictions and stores prediction history locally for offline access, thereby facilitating practical implementation in farming settings.

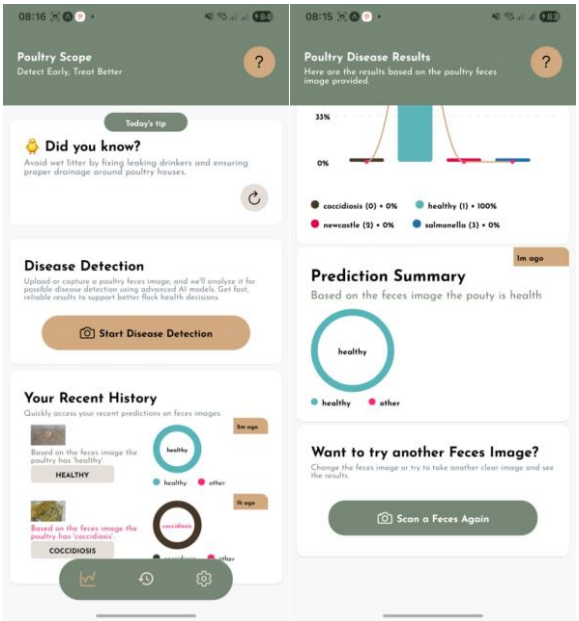


Fig. 11. "Poultry Scope" Prototype Sample Screenshots.

V. Conclusion and Future Works

This research contributes not only an accurate classification model but also a complete end-to-end deployment framework integrating MobileNetV3, FastAPI, and a native mobile interface for real-time poultry disease prediction. Unlike existing works that stop at experimental validation, this system demonstrates a deployable and scalable solution bridging model accuracy with practical usability in farm settings. This study outlines the creation and

implementation of a compact poultry disease diagnosis (PDD) solution utilizing MobileNetV3 with TL, which is integrated into a FastAPI backend and can be accessed through a native mobile application. The system demonstrated a testing accuracy of 96.07% alongside a low loss of 0.16, showcasing its robustness, reliability, and suitability for continuous operation in farm environments.

Experimental results confirm that DL models, particularly those using TL, can successfully categorize poultry faecal images into various disease classifications, enabling quick health evaluations and early diagnosis. In comparison to earlier methods, this research represents a step forward in the field by incorporating MobileNetV3 into a mobile application, decreasing inference delay while maintaining accuracy. The mobile application improves usability, giving farmers accessible, prompt diagnostic assistance, thereby aiding in the digitization of poultry health management and initiatives for food security.

Despite these encouraging outcomes, several limitations need further exploration. The dataset used was obtained from Kaggle and may not account for the environmental diversity present in various poultry farm environments, such as differences in lighting, flooring, and faecal characteristics. Expanding the dataset with geographically varied samples might enhance the model's adaptability across different farms. Furthermore, incorporating object detection frameworks like YOLO could allow for the precise localization of faecal samples in intricate, cluttered settings, thereby improving model efficiency in scenarios where mixed samples are prevalent.

Enhancing the application for offline capabilities is essential for deployment in rural areas, where access to the internet may be scarce. Future research might also examine hybrid architectures, incorporating Vision Transformers or ensemble models, which have shown encouraging outcomes in detecting poultry diseases. Lastly, extensive field trials and user studies are crucial for validating the system's usability, evaluating its influence on farm productivity, and promoting widespread acceptance among poultry farmers.

VI. Acknowledgement

The authors would like to give thanks to all individuals and institutions that contributed to the success of this research. A special thanks is extended to the Kaggle Community, more specifically [17], who are the creators of the poultry disease dataset, which served as the foundation for this study. The authors also acknowledge the broader research community whose works provided critical insights and methodological guidance that shaped this study.

VII. References

[1] S. Mugisha, R. Kisitu, and F. Tushabe, "Hybrid Knowledge Transfer through Attention and Logit Distillation for On-Device Vision Systems in Agricultural IoT," vol. 14, no. 8, pp. 1–12, 2025, [Online]. Available: http://arxiv.org/abs/2504.16128

[2] A. Monteiro, S. Santos, and P. Gonçalves, "Precision agriculture for crop and livestock farming—Brief review," *Animals*, vol. 11, no. 8, pp. 1–18, 2021, doi: 10.3390/ani11082345.

[3] G. Li, R. S. Gates, and B. C. Ramirez, "an on-Site Feces Image Classifier System for Chicken Health Assessment: a Proof of Concept," *Appl Eng Agric*, vol. 39, no. 4, pp. 417–426, 2023, doi: 10.13031/aea.15607.

[4] M. S. Hossain, U. S. Salsabil, M. M. Mahbubul Syeed, M. M. Rahman, K. Fatema, and M. F. Uddin, "SmartPoultry: Early Detection of Poultry Disease from Smartphone Captured Fecal Image," *Proceedings of JCSSE 2023 - 20th International Joint Conference on Computer Science and Software Engineering*, pp. 345–350, 2023, doi: 10.1109/JCSSE58229.2023.10202054.

[5] A.-M. Reyad, S. Dey, W. U. Bhuyian, S. I. Evan, N. Jahan, and R. Rahman, "An Ensemble Deep Learning Approach to Detect Common Chicken Diseases from Fecal Matter Images," in *Proceedings of the 3rd International Conference on Computing Advancements*, in ICCA '24. New York, NY, USA: Association for Computing Machinery, 2025, pp. 145–153. doi: 10.1145/3723178.3723198.

[6] G. Chidziwisano, E. Samikwa, and C. Daka, "Deep learning methods for poultry disease prediction using images," *Comput. Electron. Agric.*, vol. 230, no. C, Apr. 2025, doi: 10.1016/j.compag.2024.109765.

[7] G. Sichilima and J. Phiri, "Integrating Chicken Fecal Image Analysis with Machine Learning for Early Detection of Poultry Diseases in Developing Countries," *Zambia ICT Journal*, vol. 8, no. 1, pp. 24–28, 2025, doi: 10.33260/zictjournal.v8i1.337.

[8] S. M. Hassan *et al.*, "A Survey on Different Plant Diseases Detection Using Machine Learning Techniques," *Electronics (Switzerland)*, vol. 11, no. 17, pp. 1–29, 2022, doi: 10.3390/electronics11172641.

[9] M. Z. Degu and G. L. Simegn, "Smartphone based detection and classification of poultry diseases from chicken fecal images using deep learning techniques," *Smart Agricultural Technology*, vol. 4, p. 100221, Aug. 2023, doi: 10.1016/j.atech.2023.100221.

[10] H. O. Aworinde *et al.*, "Poultry fecal imagery dataset for health status prediction: A case of South-West Nigeria," *Data Brief*, vol. 50, p. 109517, 2023, doi: 10.1016/j.dib.2023.109517.

[11] D. Machuve, E. Nwankwo, N. Mduma, and J. Mbelwa, "Poultry diseases diagnostics models using deep learning," *Front Artif Intell*, vol. 5, 2022, doi: 10.3389/frai.2022.733345.

[12] A. Tasdelen and Y. Arslan, "Detection of high-risk diseases in poultry feces through transfer learning," *Engineering Science and Technology, an International Journal*, vol. 64, no. January, p. 102002, 2025, doi: 10.1016/j.jestch.2025.102002.

[13] S. Anwarul, T. Choudhury, and K. Kotecha, "DeepFowl: Disease prediction from chicken excreta images using deep learning," *Nonlinear Engineering*, vol. 14, no. 1, 2025, doi: 10.1515/nleng-2024-0081.

[14] M. N. Hasan, Z. J. Khow, V. T. Goh, S. Mansor, and Y.-F. Tan, "Image-based Detection and Classification of Poultry Diseases from Chicken Droppings in Open House Poultry Farms," *Journal of Informatics and Web Engineering*, vol. 4, no. 2, pp. 77–92, 2025, doi: 10.33093/jiwe.2025.4.2.6.

[15] B. S. Akbudak, "Disease detection from chicken feces on a mobile platform using deep learning methods," *Young Scientist Research*, vol. 6, no. 1, pp. 17–23, 2022.

[16] W. Y. Ayele, "Adapting CRISP-DM for Idea Mining A Data Mining Process for Generating Ideas using a Textual Dataset," 2020. [Online]. Available: www.ijacsa.thesai.org

[17] K. Kausthub and Plutoze, "Poultry Diseases Detection," Kaggle. Accessed: Aug. 20, 2025. [Online]. Available: https://www.kaggle.com/datasets/kausthubkannan/poultry-diseases-detection

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[19] S. Liu *et al.*, "Convolutional Normalization: Improving Deep Convolutional Network Robustness and Training," *Adv Neural Inf Process Syst*, vol. 34, no. NeurIPS, pp. 28919–28928, 2021.

[20] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[21] C. Song and A. Raghunathan, "Information Leakage in Embedding Models," *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 377–390, 2020, doi: 10.1145/3372297.3417270.

[22] S. Raschka, J. Patterson, and C. Nolet, "Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence," *Information (Switzerland)*, vol. 11, no. 4, 2020, doi: 10.3390/info11040193.

[23] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "2017(cites:1625)Random Erasing Data Augmentation," 2017.

[24] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," 2019.

[25] A. Howard, W. Wang, G. Chu, L. Chen, B. Chen, and M. Tan, "Searching for MobileNetV3 Accuracy vs MADDs vs model size," *International Conference on Computer Vision*, pp. 1314–1324, 2019.

[26] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in Vision: A Survey," *ACM Comput Surv*, vol. 54, no. 10, 2022, doi: 10.1145/3505244.

[27] W. T. Vambe and T. C. Garidzira, "A Comparative Analysis of Convolutional Neural Networks, Bi-Directional Gated Recurrent Units, and Bi-Directional Long Short-Term Memory for Sequential Data Processing Using Human Emotions Dataset," *Proceedings of 2024 4th International Multidisciplinary Information Technology and Engineering Conference, IMITEC 2024*, pp. 531–538, 2024, doi: 10.1109/IMITEC60221.2024.10851188.

[28] M. Tan and Q. V Le, "EfficientNetV2: Smaller Models and Faster Training," 2021. [Online]. Available: https://github.com/google/

[29] J. Chen, "Model Algorithm Research based on Python Fast API," *Front. Sci. Eng*, vol. 3, no. 9, pp. 7–10, 2023.

[30] A. Paleyes, R. G. Urma, and N. D. Lawrence, "Challenges in Deploying Machine Learning: A Survey of Case Studies," *ACM Comput Surv*, vol. 55, no. 6, 2023, doi: 10.1145/3533378.

[31] D. Saha, M. P. Mangukia, and A. Manickavasagan, "Real-Time Deployment of MobileNetV3 Model in Edge Computing Devices Using RGB Color Images for Varietal Classification of Chickpea," *Applied Sciences (Switzerland)*, vol. 13, no. 13, 2023, doi: 10.3390/app13137804.

[32] A. Dosovitskiy *et al.*, "an Image Is Worth 16X16 Words: Transformers for Image Recognition At Scale," *ICLR 2021 - 9th International Conference on Learning Representations*, 2021.

[33] A. Esteva *et al.*, "A guide to deep learning in healthcare," *Nat Med*, vol. 25, no. 1, pp. 24–29, 2019, doi: 10.1038/s41591-018-0316-z.

[34] Z.Diko and K.Sibanda, 2024. Comparative Analysis of Popular Supervised Machine Learning Algorithms for Detecting Malicious Universal Resource Locators, Journal of Cyber Security and Mobility, Vol. 135,1105–1128.doi: 10.13052/jcsm2245-1439.13513, © 2024 River Publishers. doi: 10.13052/jcsm2245-1439.13513

[35] L. Mabandla and K. Sibanda, 2022. A Comparative Analysis of Light Weight Intrusion Detection Models for the Internet of Things, Southern Africa Telecommunication Networks and Applications Conference (SATNAC), ISBN: 978-0-6397-2773-8.

WhaleVAD-BPN

Improving Baleen Whale Call Detection with Boundary Proposal Networks and Post-processing Optimisation

Christiaan M. Geldenhuys cmgeldenhuys@sun.ac.za
Günther Tonitz gtonitz@sun.ac.za
Thomas R. Niesler trn@sun.ac.za

Department of Electrical and Electronic Engineering
University of Stellenbosch
Stellenbosch, South Africa

***Abstract*—While recent sound event detection (SED) systems can identify baleen whale calls in marine audio, challenges related to false positive and minority-class detection persist. We propose the boundary proposal network (BPN), which extends an existing lightweight SED system. The BPN is inspired by work in image object detection and aims to reduce the number of false positive detections. It achieves this by using intermediate latent representations computed within the backbone classification model to gate the final output. When added to an existing SED system, the BPN achieves a 16.8 % absolute increase in precision, as well as 21.3 % and 9.4 % improvements in the F1-score for minority-class d-calls and bp-calls, respectively. We further consider two approaches to the selection of post-processing hyperparameters: a forward-search and a backward-search. By separately optimising event-level and frame-level hyperparameters, these two approaches lead to considerable performance improvements over parameters selected using empirical methods. The complete WhaleVAD-BPN system achieves a cross-validated development F1-score of 0.475, which is a 9.8 % absolute improvement over the baseline.**

***Index Terms*—Sound Event Detection, Computational Bioacoustics, Baleen Whale Call Detection, Marine Bioacoustics, Boundary Proposal Network, Post-Processing**

I. Introduction

Passive acoustic monitoring (PAM) has become a cornerstone for assessing marine mammal populations in remote and otherwise inaccessible habitats, owing to its non-invasive nature and relatively modest financial cost. In practice, however, PAM generates substantial volumes of data, often suffering from low signal-to-noise ratios in recordings [1]. Both factors render manual annotation labour-intensive and dependent on specialist knowledge. Consequently, a large body of research has focused on the development of automated detectors for the vocalisations of key species, most notably *Balaenoptera musculus* (blue) and *B. physalus* (fin) whales. These species remain classified as endangered and vulnerable by the IUCN [2], [3], and continue to elude precise abundance estimates because of a scarcity of labelled data [4].

Recent sound event detection (SED) systems have made considerable strides in identifying baleen whale calls within marine recordings, yet these models still exhibit a high false-positive rate and do not allow the reliable detection of infrequent (minority-class) call events [5]. To address these shortcomings, we propose the inclusion of a complementary neural module, the boundary proposal network (BPN). Inspired by the field of object detection in image processing [6], [7], the BPN is intended to supplement an existing lightweight SED architecture by exploiting intermediate latent features already calculated within the backbone classifier. The BPN output is used as a gating mechanism that refines the temporal localisation of detected events, thereby reducing false positives and improving overall precision. It was found that the BPN also leads to an improvement in recall for minority-class call types.

In addition to the described architectural augmentation, we introduce improvements to the post-processing stage of the SED architecture itself. We investigate two search strategies for optimising the hyperparameter selection of the post-processing stage: a forward-search and a backward-search. These procedures systematically explore event- and frame-level hyperparameters, yielding performance gains that surpass those obtained through existing ad hoc or empirical methods.

II. Background

This section provides a summary of the background regarding baleen whale call detection, proposal networks, and the post-processing techniques used.

A. Baleen whale call activity detection

Baleen whales (*mysticetes*) undertake extensive migrations and thus communicate using low-frequency vocalisations that propagate over great distances underwater. Previous work has shown that voice activity detection (VAD) algorithms can be applied to baleen whale call detection, specifically detecting the calls of blue and fin whales [5]. For example, the AVA-VAD [8] system relied on producing latent features from a spectrogram representation of the audio recording, using a convolutional neural network (CNN). These latent features are then processed sequentially by a bidirectional long short-term memory network (BiLSTM) architecture with sigmoid activation to obtain the final posterior classification probabilities for each frame in the input spectrogram.

While phase information has been disregarded in speech processing systems, its inclusion has been reported to afford a $10\,\%$ improvement in the F1-score [5]. This work proposed

further changes to the AVA-VAD architecture, with the addition of bottleneck and depthwise convolutional layers with recurrent connections. The final proposed system was called WhaleVAD, and it is this system that we will extend.

B. Proposal networks

Proposal networks are a two-stage architecture common in the computer vision field for achieving object detection [6], [9], [10]. These networks have been refined over time to be computationally efficient and have proven to be effective at localising and identifying objects within an image. Early approaches, such as R-CNN (regions with CNN features) [9], relied on external algorithms, such as *selective search* [11], to generate a set of candidate regions of interest (ROIs). These ROIs could then be used by a secondary detection network to determine if a particular object is present in the proposed region. However, these proposal networks relied on handcrafted convolutional features.

Ren, He, Girshick, *et al.* [6] were the first to introduce the region proposal network (RPN), which integrated proposal generation directly into the neural network architecture. The technique relied on applying a small trainable CNN to the feature map of a convolutional backbone network to coordinate and refine a set of predefined anchor boxes. By sharing convolutional features with the downstream detection network, the RPN enabled an efficient and end-to-end trainable system. This architecture established the foundation for most subsequent two-stage object detectors, demonstrating the effectiveness of backbone features.

Despite the success of RPNs, its reliance on a single, deep feature map presents limitations in detecting objects across varying scales and remains computationally less efficient when compared to single-stage detectors. Liu, Anguelov, Erhan, *et al.* [7] address this by attaching multiple detection heads to intermediate layers at varying depths within the backbone network. Other single-stage detectors, such as YOLOv3 [12], adopted a multi-scale feature pyramid network-inspired (FPN) [13] mechanism to aggregate intermediate features.

C. Post-processing

During per-frame SED in a bioacoustic system, the classification model produces a sequence of probabilities over time, each indicating the likelihood of a particular sound class being present in a frame. These *model probabilities* are converted to *binary detections* by applying a threshold to each of the per-frame call probabilities. These binary detections are then aggregated into *discrete events* representing contiguous periods of (call) activity.

The ideal system output is a single event that accurately aligns with the human annotation. In practice, however, these output events are often fragmented, containing intermittent gaps, or spurious or excessively long detections.

To mitigate this, filtering is applied during post-processing. We consider two classes of filtering techniques, namely frame- and event-level.

1) Frame-level techniques: These methods serve to smooth either the model output probabilities, or the detections, or both. For example, a median filter can be applied to the model output, where each probability is replaced by the median value within a neighbourhood. This reduces sporadic peaks or dips in activity, thereby decreasing the number of fragmented events, at the cost of reduced precision. Hysteresis is another common frame-level post-processing approach, where a window of past estimates influences the current model decision. One implementation applies different thresholds for entering and exiting an event state, which we will refer to as *threshold hysteresis* [14]. By setting a lower threshold for termination, the system remains active for longer, which may help reduce event fragmentation. Alternatively, activity at a given time instant can be defined based on the majority vote (statistical mode) within the sliding window, which we will refer to as *hangover*. This can be interpreted as a variant of median filtering applied to model binary detections, as opposed to model class probabilities; given by the following equation:

$$\tilde{y}_t = \begin{cases} 1 & \text{if } \sum_{i=0}^{k} \hat{y}_{t-i} > \dfrac{k+1}{2}, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $\tilde{y}_t$ is the model detection at time instant t, and k is the number of past samples in the sliding window.

2) Event-level techniques: These methods act at an event-level on the aggregated detections and commonly impose constraints on duration. Minimum and maximum event durations may be defined to remove events of implausible duration, and a minimum inter-event duration may be defined to merge events occurring close in time.

The effectiveness of these post-processing methods depends on a set of hyperparameters, such as the size of the neighbourhood used to implement median filtering or hangover. The selection of these hyperparameters is highly task and data-specific, and thus requires optimisation.

III. Literature Review

This section starts with an overview of previous research in the detection and classification of baleen whales, followed by a focus on post-processing strategies that have been implemented and evaluated in bioacoustic SED.

A. Detection and classification of baleen whale calls

Automated methods for the detection and classification of baleen whales continue to develop. Due to the great distances over which their vocalisations propagate underwater, these calls are an ideal candidate for PAM [15].

Before the advent of machine learning, automated detection relied on classical signal-processing techniques designed to identify signals with known, empirical characteristics. Two common approaches were matched filtering, in which a synthetic kernel derived from a known call is correlated with a recording to locate vocalisations in background noise [15], and spectrogram correlation, which cross-correlates a template

spectrogram with successive recording segments to identify vocal occurrences [16].

Machine learning approaches were subsequently adopted to address the non-stationary nature of ocean soundscapes and reduce the reliance on fixed filters or templates. Examples include support vector machines (SVMs) to identify individual humpback whale vocalisations [17] and North Atlantic right whale upcalls [18], as well as probabilistic models such as Gaussian mixture models (GMMs) for blue whale calls [19] and hidden Markov models (HMMs) for humpback whale calls [20].

More recently, deep neural networks (DNNs), particularly CNNs, have become a common area of research for computational bioacoustics. Large CNN architectures, such as DenseNet [21], have been applied to baleen whale classification; for example, Miller, Madhusudhana, Aulich, *et al.* [22] reported that a DenseNet-based system surpassed human observers in blue whale D-call detections. Convolutional recurrent neural networks (CRNNs), which integrate recurrent layers to capture temporal dependencies, have also shown strong performance in fin- and blue whale classification [23]. Despite continued improvements in frame-level detector performance [24], lower false-positive rates are desirable, as frame-level errors can cascade into greater event-level errors.

B. Post-processing in bioacoustic SED

Frame-level post-processing commonly involves class-dependent thresholds [4], [25], [26] and median filters with a fixed kernel size [26]. Hoffman and Robinson [25] report that class-specific thresholds, applied to logits, can outperform a fixed threshold on bird-call detection. Hysteresis thresholds have not been reported in the bioacoustic literature. However, this technique has been applied in other SED domains. For example, Cances, Pellegrini, and Guyot [27] found hysteresis thresholds to outperform absolute thresholds when classifying sound classes that occur in a domestic environment. Hangover has also not been reported on in SED studies, although it has been applied successfully in other detection problems, such as sonar [28] and radar [29], where an *M-out-of-N* criterion is used.

Event-level post-processing techniques are common in few-shot bioacoustic learning. For example, several studies impose a minimum event duration and a minimum inter-event interval. These hyperparameters are commonly derived from the support set [25], [30]. Miller *et al.* [4] apply a minimum and maximum event duration of $0.5\,\mathrm{s}$ and $2.5\,\mathrm{s}$ respectively, and require at least $0.5\,\mathrm{s}$ between events when identifying fin whale $20\,\mathrm{Hz}$ pulses.

For domestic environmental SED, Cances, Guyot, and Pellegrini [14] have provided a comprehensive comparison of frame-level post-processing techniques, including fixed threshold, hysteresis threshold, and a slope-based threshold technique which detects fast changes in model probabilities. Each of the threshold techniques were evaluated using class-dependent and class-independent hyperparameters. The authors report an $28.6\,\%$ absolute improvement in the F1-score over the baseline, when selecting class-dependent fixed thresholds. This suggests that the selection of class-specific hyperparameters and the use of multiple post-processing strategies in bioacoustic SED may yield substantial gains in detection performance. However, to our knowledge, no bioacoustic SED study has reported a systematic treatment of different post-processing strategies.

IV. Data

As part of the 2025 IEEE BioDCASE (Task 2) challenge, a dataset consisting of strongly labelled blue- and fin-whale calls in the Antarctic region of the Southern Ocean was released [31]. The data was originally obtained by the Antarctic Blue and Fin Whale Acoustic Trends Project (ATP) as part of the International Whaling Commission's Southern Ocean Research Partnership (IWC-SORP) [32].

The Acoustic Trends Blue Fin Library (ATBFL) consists of 11 site-year datasets recorded around the Antarctic, in the period 2005 to 2017. Sites were selected based on the availability of a full year of data, and utilised different recording instrumentation. All audio data was resampled to a rate of $250\,\mathrm{Hz}$. Each dataset was manually annotated, in both the time and frequency domains, by domain experts using the data collection and annotation procedures described in Miller *et al.* [4].

The challenge identifies three site-year datasets as a development set, while the remaining eight site-year sets form the training set. The entire dataset contains a total of $6594\,\mathrm{recordings}$ with a total duration of $76\,178\,\mathrm{hours}$, as set out in Table I. Only $5.2\,\%$ of the data (in duration) contains blue or fin whale vocalisations, however.

The ATBFL includes annotations for seven different call types, of which four are produced by blue whales (*BmA*, *BmB*, *BmZ* and *BmD*) and three by fin whales (*BpD*, *Bp20* and *Bp20plus*). A blue whale Z-call (*BmZ*) is a low-frequency compounded call consisting of both the A-call (*BmA*) and the B-call (*BmB*). The blue whale D-call (*BmD*) is a downsweeping call, ranging between $20\,\mathrm{Hz}$ to $120\,\mathrm{Hz}$, which is similar to a fin whale downsweep (*BpD*) which ranges from $30\,\mathrm{Hz}$ to $90\,\mathrm{Hz}$. Finally, fin whales also create a downsweeping pulse between $15\,\mathrm{Hz}$ to $30\,\mathrm{Hz}$ which can either appear with (*Bp20plus*) or without (*Bp20*) an overtone varying between $80\,\mathrm{Hz}$ to $120\,\mathrm{Hz}$. Figure 1 shows examples of each call type.

Following Schall, Kaya, Debusschere, *et al.* [24], call types are grouped based on their acoustic similarity and interrelated usage. The A-, B-, and Z-calls, which co-occur, are combined into a single ABZ-call category (`bmabz`). Similarly, the blue whale D-call and the fin whale BpD-call are grouped into a unified D-call category (`d`), while the Bp20 and Bp20Plus calls are merged into the Bp-call category (`bp`). These call grouped categories serve as the final call labels used.

Table II summarises the duration, frequency characteristics, and annotation counts for all call types found in the training and the development sets. Blue whale vocalisations are more common in both datasets, comprising $74.8\,\%$ of the training set and $65.9\,\%$ of the development set annotations. Notably,

TABLE I: Summary of the ATBFL site-year training and development sets. The table shows the average recording duration (hours), number of recordings, total recording duration (hours), number of annotated events, and total duration of whale calls (hours) for each set.

Dataset	Avg. Duration (h)	Recordings	Total duration (h)	Total events	Total event duration (h)
ballenyisland2015	1.0	205	204	2222	2.8
casey2014	1.0	194	194	6866	14.2
elephantislands2013	0.08	2247	187	21949	16.1
elephantislands2014	0.08	2595	216	20962	28.1
greenwich2015	0.17	190	32	1128	2.1
kerguelen2005	1.0	200	200	2960	3.5
maudrise2014	0.42	200	83	2360	5.7
rosssea2014	1.0	176	176	104	0.1
Total training set	–	**6007**	**1292**	**58551**	**72.6**
casey2017	1.0	187	185	3263	6.1
kerguelen2014	1.0	200	200	8822	11.4
kerguelen2015	1.0	200	200	5542	7.4
Total development set	–	**587**	**585**	**17627**	**24.9**

TABLE II: Different whale call frequency (hertz) and duration (seconds) information computed from data in [31].

(a) Training Set

Type	Frequency (Hz)			Duration (s)			Count
	Min	Max	Avg.	Min	Max	Avg.	
BmA	11.4	110.6	25.9	2.12	27.11	7.19	13785
BmB	10.0	31.3	22.2	3.14	19.51	7.83	5433
BmZ	11.5	34.6	22.0	3.87	28.07	12.76	1646
BmD	11.5	110.7	69.9	0.29	6.78	1.42	22977
BpD	16.7	134.1	75.4	0.29	2.70	1.12	2658
Bp20	8.5	45.1	22.2	0.48	3.08	1.52	9104
Bp20plus	9.2	112.7	52.5	0.76	2.91	1.50	3950

(b) Development Set

Type	Frequency (Hz)			Duration (s)			Count
	Min	Max	Avg.	Min	Max	Avg.	
BmA	15.7	30.1	25.6	2.12	36.62	7.12	6268
BmB	10.7	99.0	22.6	1.29	18.1	8.35	2277
BmZ	12.1	30.3	21.9	5.15	29.45	12.64	918
BmD	15.9	122.9	57.4	0.74	7.36	2.87	2168
BpD	26.5	137.5	61.7	0.37	2.58	1.08	688
Bp20	10.3	47.9	22.7	0.46	2.83	1.35	2550
Bp20plus	11.1	106.6	57.1	0.64	2.58	1.43	2758

blue whale A-calls, B-calls, and Z-calls exhibit longer call durations, accounting for $45.18\,\text{hours}$ in the training set and $20.9\,\text{hours}$ in the development set. This distribution reveals an imbalance, both in species representation (blue whales are overrepresented) and in the temporal occurrence of call types.

V. Experimental Structure

In the following, we first describe the architectural changes we have made to the WhaleVAD system. Then, we present our boundary proposal network, which used intermediate features from within the WhaleVAD system to improve overall model performance. Finally, we present two post-processing optimisation search strategies that separately optimise frame-level and event-level hyperparameters.

A. Architecture modifications

In the original WhaleVAD model [5], the *depthwise convolution* block consisted of three depthwise convolution layers placed in series. We adapt the depthwise convolution block by adding residual connections between each of the layers with increasing dilation of 2, 4 and 8. The increase in dilation factor provides a wider receptive field, thus allowing the network to utilise features that are further away in time [33]. Each depthwise convolution block retains the GELU activation and batch normalisation used by the original model. In addition, the conventional dropout has been replaced with spatial dropout [34], which has been shown to improve the effective regularisation for convolutional layers.

B. Boundary proposal network

While the original WhaleVAD model architecture achieved high recall, the model also exhibited a high false positive rate (FPR), which in turn resulted in a low precision [5]. We propose a new component, the boundary proposal network (BPN), whose purpose is to compute a gating score that is combined with the WhaleVAD classifier output in an effort to reduce false positives. The boundary proposal network (BPN) uses convolutional feature representations obtained from intermediate layers, referred to as *intermediate feature maps*, from the backbone WhaleVAD classifier, to compute this gating score.

The following subsections describe the components that comprise the BPN, as well as the training regime used.

1) Intermediate projection head: Each set of intermediate feature maps is processed by a separate CNN, called an intermediate projection head. This consists of a convolutional layer with batch normalisation, GELU activation and a final maximum pooling layer. Each projection head shares the same architecture, but has its own set of weights associated with a particular point in the backbone network, from which the feature map was drawn.

2) Proposal network: The outputs of all heads are concatenated along a new dimension H, which is processed by the

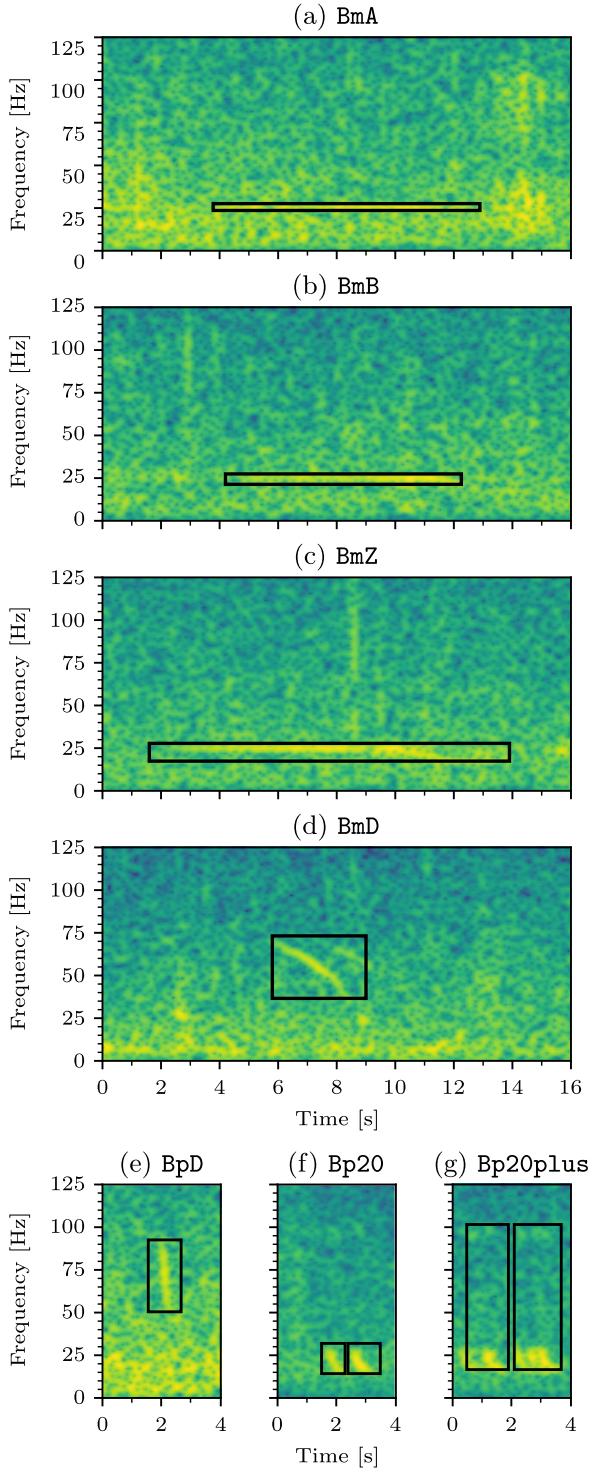


Fig. 1: Spectrogram representations of exemplar baleen whale call types [31]. Bounding boxes indicate presence of particular call type, provided by domain expert annotations in dataset. Figures (a) – (d) correspond to blue whale vocalisations, and (e) – (g) correspond to fin whale vocalisations.

TABLE III: The boundary proposal network (BPN) layerwise configuration in terms of kernel size (K), stride (S), number of input channels (C_{in}) and output channels (C_{out}).

Layer	K	S	C_{in}	C_{out}
Intermediate projection head				
└ Conv2D	(1, 1)	(1, 1)	128	128
└ Max pool	(3, 1)	(1, 1)	–	–
Proposal network				
└ Conv2D transpose	(4, 1)	(1, 1)	128	128
└ Conv2D transpose	(5, 1)	(1, 1)	128	64

proposal network to produce R distinct ROI vectors per head. Each ROI consists of a latent feature vector with dimensionality C_{bpn}, associated with a projection head ($h_i \in H$) for each time instant T. The proposal network consists of two convolutional layers with GELU activation, batch normalisation and spatial dropout [34]. We evaluate two variants of the network. *BPN-multi* produces multiple ROIs per projection head using a convolutional transpose ($R > 1$), whilst *BPN-single* produces a single ROI per projection head ($R = 1$). Early experimentation showed that BPN-multi outperformed BPN-single and thus only BPN-multi was considered in the final results, and will be referred to as simply BPN throughout. Table III shows a summary of the BPN layer configuration.

3) BiLSTM: Each latent ROI vector is processed sequentially in time by a BiLSTM or independently by a logistic regression (LR) module; with a sigmoid activation on the output. During hyperparameter optimisation, it was found that the BiLSTM network outperforms LR. The resulting output is averaged over each ROI using a learned weighted mean to produce a *mask*. The weighted average is jointly trained with the model, resulting in some heads being weighted more heavily than others. This weighting remains fixed during inference.

4) Masking: The final mask is applied to the posterior call probabilities produced by the backbone classifier, thus acting as a soft gating mechanism whose purpose is to suppress spurious detections (false positives). As a result, the final posterior call probabilities are dependent on both the classification network, which is responsible for localising and identifying a particular call type in time, and the gating mechanism of the BPN which aligns with this postulated call. The addition of the BPN gating mechanism should therefore allow the number of false positive classifications made by the original WhaleVAD architecture to be reduced through training. We will refer to our model as WhaleVAD-BPN. Figure 2 shows an illustration of the complete WhaleVAD-BPN system.

5) Training regime: All models were trained using the AdamW [35] optimiser with Focal loss [36]. The training set was divided into mini-batches of 48 segments per batch, each consisting of approximately 30 seconds long. The learning rate is kept fixed at 0.001 with momentum terms of 0.9 and 0.999 and a weight decay factor of 0.01. Training is halted once the training loss has converged or after 32 epochs over the entire training set.

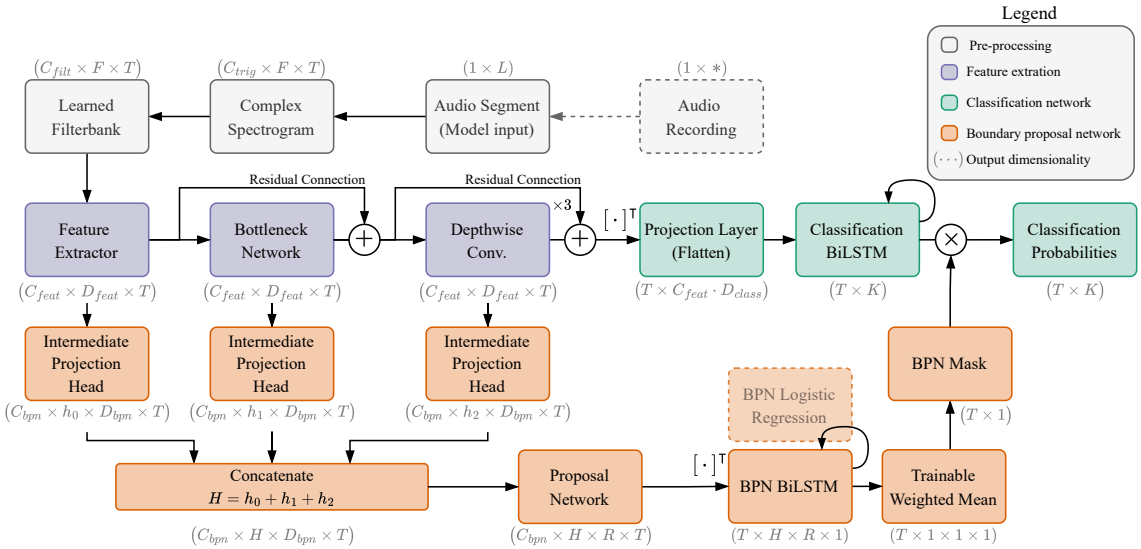


Fig. 2: Illustration of the WhaleVAD-BPN system we propose for improved whale call boundary detection. The system is divided into four sections: pre-processing, feature extraction, classification, and boundary proposal. The output tensor dimensionality is shown below each module in the system.

C. Post-processing hyperparameter selection

During the evaluation of the different post-processing techniques, we employ a three-fold cross-validation scheme using the development sets [37]. For cross-validation, the data is partitioned into disjoint sets, referred to as *folds*. The development set consists of three site-year subsets, each of which is assigned to a different fold. The best post-processing hyperparameters are chosen based on the highest F1-score over two of the folds (development folds), while the third fold is held out for testing (test fold). After the parameters are chosen based on the development folds, the system is evaluated on the test fold. The chosen development and test folds are permuted, referred to as a *turn*, and the process is repeated. The final system evaluation is computed by averaging the score over all three turns of the test folds.

Each post-processing hyperparameter is either applied at a frame-level or at an event-level, as discussed in Section II-C. We propose two selection procedures, namely *forward-search* and *backward-search*. During either search method, both frame-level and event-level hyperparameters are searched separately using a two-stage process. An exhaustive optimisation across both hyperparameter types was practically infeasible, as it would require either substantial computational resources or a large reduction in the search space.

1) Forward-search: During the forward-search (refer to Fig. 3b), event-level hyperparameters are initially fixed (Stage 1) based on the statistical properties derived from the dataset. The minimum inter-event duration is fixed at 500 ms. For each of the three target classes, the minimum and maximum event durations are drawn from the ranges in Table II by taking, within each group, the overall minimum and maximum of the constituent classes. The frame-level hyperparameters are searched based on Table IVa: median filter kernel size,

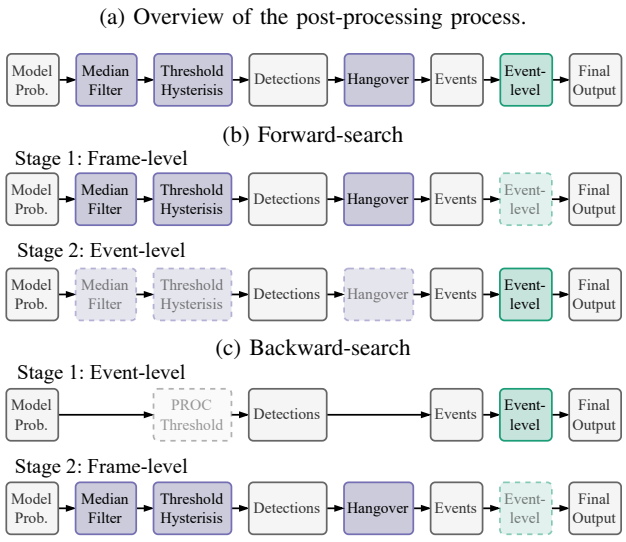


Fig. 3: Illustration of the post-processing applied to the call probabilities produced by a per-frame SED model. (a) An overview of the steps employed during both frame-level (*purple*) and event-level (*green*) post-processing. (b and c) The forward- and backward-search approaches to the selection of post-processing hyperparameters, respectively. Both optimisation methods consist of two stages, where some parts of the network remain fixed (opaque dashes) while hyperparameters for the remainder are searched to maximise the F1-score over all development folds.

TABLE IV: Ranges of the hyperparameter values considered for (a) frame-level, and (b) event-level post-processing.

(a) Frame-level hyperparameters.

Parameter	Search space
Filter kernel	[None, 11, 33, 55]
On threshold	$0.1 \rightarrow 0.9$ (inc. 0.1)
Off threshold	$0.1 \rightarrow 0.9$ (inc. 0.1)
Hangover kernel	[None, 11, 33, 55]

(b) Event-level hyperparameters.

Parameter	bmabz	d	bp
Min. time between events (s)	$0.1 \rightarrow 0.9$ (inc. 0.1)	$0.1 \rightarrow 0.9$ (inc. 0.1)	$0.1 \rightarrow 0.9$ (inc. 0.1)
Min. event duration (s)	$2.0 \rightarrow 5.0$ (inc. 0.5)	$0.6 \rightarrow 3.0$ (inc. 0.4)	$0.3 \rightarrow 1.5$ (inc. 0.2)
Max. event duration (s)	$25.0 \rightarrow 40.0$ (inc. 2.5)	$5.0 \rightarrow 11.0$ (inc. 1.0)	$2.0 \rightarrow 5.0$ (inc. 0.5)

threshold, hysteresis (off) threshold, and the hangover kernel size. Note that each of these hyperparameters is class-dependent. Next, in Stage 2, these candidate frame-level hyperparameters are fixed, while the event-level hyperparameters are searched based on Table IVb.

2) Backward-search: During the backward-search (refer to Fig. 3c), frame-level hyperparameters (thresholds) are initially fixed (Stage 1) based on equal precision-recall, obtained from the average precision-recall curve computed over the development folds. Event-level hyperparameters are searched based on Table IVb. Next, in Stage 2, these candidate event-level hyperparameters are fixed, while the frame-level hyperparameters are considered based on Table IVa.

3) Final model evaluation: After the hyperparameters have been fixed using either search method, precision and recall metrics are computed for each test fold. The final F1-score, per call type, is then recomputed from the averaged precision/recall scores. The final system evaluation is based on the macro F1-score of each class, again using the averaged precision/recall scores [31]. The hyperparameter selection process is repeated for both the WhaleVAD and the WhaleVAD-BPN models.

VI. Results

Table V shows the cross-validated performance of the baseline model when applying hyperparameter selection, using backward-search, on a per-class basis. Applying event-level optimisation alone led to consistent performance gains for all classes, when compared to the previously-used empirical method of selecting the post-processing parameters (Table VI). In particular, for D-calls (d) we see a $9.3\,\%$ absolute improvement in the F1-score, which corresponds to a $77.5\,\%$ relative improvement over the empirically derived parameters. Additional frame-level optimisation provides more modest gains, leading to a final macro F1-score of 0.422 across all classes. Overall, this represents a $4.5\,\%$ absolute improvement in the macro F1-score, compared to the baseline WhaleVAD model, achieved solely through the optimisation of post-processing and no architectural modifications (see Table VII). Forward-search achieved similar performance to backward-search. However, since backward-search performed slightly better, results for only this method will be shown.

Figure 4 shows that the WhaleVAD-BPN architecture yields superior precision-recall curves across all three call types relative to the original WhaleVAD system, indicating better performance even without optimisation of the post-processing. The model successfully leads to improved precision (fewer false positives) across a broad range of classification thresholds and particularly for the minority classes d and bp.

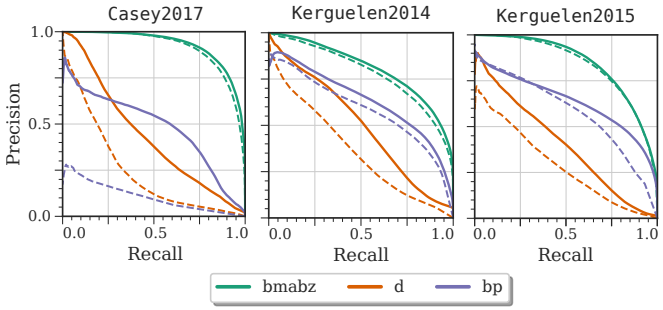


Fig. 4: Precision-recall curves for each call type and for each of the three site-year development sets. Both the baseline WhaleVAD (dashed) and the proposed WhaleVAD-BPN (solid) models are shown.

The application of hyperparameter optimisation using backward-search for the proposed WhaleVAD-BPN model is shown in Table V. Frame-level optimisation achieves substantial improvements and is more successful than event-level optimisation. Specifically, bmabz-, d- and bp-calls improve by $6.9\,\%$, $9.8\,\%$, and $4.8\,\%$, respectively. The final macro F1-score reaches 0.475, representing a $9.8\,\%$ absolute improvement over the original WhaleVAD model and a $5.3\,\%$ improvement over WhaleVAD with optimised post-processing (see Table VII).

To investigate the individual contribution of each frame-level post-processing technique, we also evaluated the absolute improvement when each technique is applied in addition to selecting class-dependent thresholds. Specifically, we consider class-dependent hysteresis thresholds, hangover, and median filtering. Although a few cases yielded noticeable gains, most contributed relatively little compared to class-dependent threshold optimisation alone. As shown in Table V, event-level hyperparameter optimisation produces large performance gains, challenging the prevalent current practice of simply deriving these values from dataset statistics.

Finally, Table VII highlights the trade-offs underlying these gains. The optimised WhaleVAD model improves the F1-score primarily by sacrificing recall in exchange for higher precision. By contrast, WhaleVAD-BPN manages to achieve a marginal improvement in recall compared to the baseline WhaleVAD model whilst also achieving a substantial increase in precision. Therefore, the boundary proposal network has succeeded in reducing false positives while preserving recall.

TABLE V: Cross-validation F1-score for WhaleVAD (baseline) and WhaleVAD-BPN models using a backward-search for selecting the post-processing hyperparameters. (a, Stage 1) Event-level: classification thresholds are selected from the average precision-recall curve over the development folds, after which event-level hyperparameters are selected. (b, Stage 2) Frame-level: hyperparameters are selected, while event-level hyperparameters remain fixed at the previously selected values. The final F1-score is recalculated based on the average of the precision and recall for each test fold.

(a) Stage 1: Event-level selection

Call type	Fold 1		Fold 2		Fold 3		Final F1
	Dev	Test	Dev	Test	Dev	Test	
WhaleVAD							
bmabz	0.634	0.686	0.665	0.620	0.670	0.591	**0.663**
d	0.212	0.162	0.226	0.186	0.199	0.220	0.219
bp	0.529	0.029	0.291	0.446	0.237	0.560	0.348
WhaleVAD-BPN							
bmabz	0.546	0.510	0.536	0.644	0.526	0.410	0.546
d	0.238	0.233	0.204	0.152	0.235	0.307	**0.242**
bp	0.512	0.288	0.376	0.451	0.339	0.489	**0.412**

(b) Stage 2: Frame-level selection

Call type	Fold 1		Fold 2		Fold 3		Final F1
	Dev	Test	Dev	Test	Dev	Test	
WhaleVAD							
bmabz	0.626	0.698	0.659	0.632	0.665	0.621	**0.669**
d	0.209	0.223	0.233	0.175	0.199	0.242	0.222
bp	0.530	0.029	0.302	0.485	0.257	0.575	0.363
WhaleVAD-BPN							
bmabz	0.590	0.603	0.585	0.614	0.608	0.567	0.615
d	0.311	0.388	0.377	0.256	0.322	0.366	**0.340**
bp	0.532	0.311	0.442	0.492	0.401	0.573	**0.460**

TABLE VI: Cross-validation F1-score for WhaleVAD when no frame-level post-processing is applied and event-level hyperparameters are selected from empirical call statistics.

Type	Fold 1		Fold 2		Fold 3		Final F1
	Dev	Test	Dev	Test	Dev	Test	
bmabz	0.615	0.620	0.620	0.657	0.631	0.536	0.628
d	0.136	0.049	0.106	0.128	0.099	0.168	0.126
bp	0.494	0.038	0.286	0.386	0.217	0.521	0.315

TABLE VII: Final results comparing the original WhaleVAD architecture, a variant with post-processing optimised via cross-validation, and the best reported model, WhaleVAD-BPN, also with optimisation. Scores are averaged across call types and test folds.

Model	IoU	Recall	Precision	F1
WhaleVAD [5]	0.588	0.458	0.320	0.377
WhaleVAD + optimisation	0.601	0.391	0.458	0.422
WhaleVAD-BPN + optimisation	**0.625**	**0.463**	**0.488**	**0.475**

VII. Conclusion

In this work, we present a novel and computationally lightweight network augmentation (BPN) for an existing whale call detection system, as well as a computationally tractable approach to hyperparameter selection for the system post-processing. Both innovations are shown to lead to substantial performance improvements over the baseline system using the 2025 DCASE (Task 2) challenge data, for automated baleen whale call detection.

By exploiting intermediate latent features already computed within the main classifier to act as a gating mechanism for the output, the BPN consistently reduces false positive rates across all classes. The augmented architectures have also shown to improve minority-class call detection, which is generally more difficult than the detection of abundant classes. This is important because annotated data is difficult to obtain, and therefore better performance for a small pool of training examples is highly desirable.

We further demonstrate that the principled selection of post-processing hyperparameters has a marked impact on final system performance. We compare two hyperparameter selection strategies, namely a forward- and backward-search, which both achieve comparable gains. When comparing hyperparameters selected in this way to conventional empirical or ad-hoc choices, a $4.5\,\%$ absolute improvement is seen.

Our final system, which includes the proposed BPN and optimised post-processing hyperparameters, achieves a $9.8\,\%$ absolute improvement in overall F1-score. The system succeeds in markedly reducing false positives, while improving the detection of minority-class calls. In addition to maintaining the already strong recall performance of the baseline baleen whale call detection system, these improvements narrow the performance gap between minority-class calls and calls for which there is abundant data. The proposed system should therefore be a useful tool for the discovery and monitoring of new call types, for which data will initially always be limited.

Acknowledge

The authors gratefully acknowledges Telkom (South Africa) for their financial support and thanks the Stellenbosch Rhasatsha high performance computing (HPC1) facility for the compute time provided to the research presented in this paper. The authors also acknowledges the anonymous reviewers for their insightful comments, which have improved the clarity of this manuscript.

References

[1] K. A. Kowarski and H. Moors-Murphy, "A review of big data analysis methods for baleen whale passive acoustic monitoring," *Marine Mammal Science*, vol. 37, no. 2, pp. 652–673, 2021, ISSN: 0824-0469, 1748-7692. DOI: 10.1111/mms.12758.

[2] J. G. Cooke, "Balaenoptera physalus," *The IUCN Red List of Threatened Species*, 2018.

[3] J. G. Cooke, "Balaenoptera musculus," *The IUCN Red List of Threatened Species*, 2018, Erratum published in 2019.

[4] B. S. Miller, The IWC-SORP/SOOS Acoustic Trends Working Group, K. M. Stafford, *et al.*, "An open access dataset for developing automated detectors of antarctic baleen whale sounds and performance evaluation of two commonly used detectors," *Scientific Reports*, vol. 11, no. 1, p. 806, 2021, ISSN: 2045-2322. DOI: 10.1038/s41598-020-78995-8.

[5] C. M. Geldenhuys, G. Tonitz, and T. R. Niesler, "Whale-VAD: Whale vocalisation activity detection," DCASE2025 Challenge, Technical Report, 2025.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-CNN: Towards real-time object detection with region proposal networks," in *Proceedings of Advances in Neural Information Processing Systems (NIPS 2015)*, vol. 28, Montreal, Canada, 2015.

[7] W. Liu, D. Anguelov, D. Erhan, *et al.*, "SSD: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, Netherlands: Springer, 2016, pp. 21–37.

[8] N. Wilkinson and T. Niesler, "A hybrid CNN-BiLSTM voice activity detector," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, Canada: IEEE, 2021, pp. 6803–6807, ISBN: 978-1-72817-605-5. DOI: 10.1109/ICASSP39728.2021.9415081.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[10] R. Girshick, "Fast r-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[11] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013, ISSN: 1573-1405. DOI: 10.1007/s11263-013-0620-5.

[12] J. Redmon and A. Farhadi, *YOLOv3: An incremental improvement*, 2018. DOI: 10.48550/arXiv.1804.02767. arXiv: 1804.02767[cs].

[13] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, 2017, pp. 2117–2125.

[14] L. Cances, P. Guyot, and T. Pellegrini, *Evaluation of post-processing algorithms for polyphonic sound event detection*, 2019. DOI: 10.48550/arXiv.1906.06909.

[15] D. K. Mellinger and C. W. Clark, "Methods for automatic detection of mysticete sounds," *Marine and Freshwater Behaviour and Physiology*, vol. 29, pp. 163–181, 1997.

[16] D. K. Mellinger and C. Clark, "Recognizing transient low-frequency whale sounds by spectrogram correlation.," *The Journal of the Acoustical Society of America*, vol. 107, no. 6, pp. 3518–29, 2000. DOI: 10.1121/1.429434.

[17] S. Mazhar, T. Ura, and R. Bahl, "Vocalization based individual classification of humpback whales using support vector machine," in *OCEANS 2007*, 2007, pp. 1–9. DOI: 10.1109/OCEANS.2007.4449356.

[18] A. K. Ibrahim, H. Zhuang, N. Erdol, and A. M. Ali, "A new approach for north atlantic right whale upcall detection," in *2016 International Symposium on Computer, Consumer and Control (IS3C)*, 2016, pp. 260–263. DOI: 10.1109/IS3C.2016.76.

[19] A. Cuevas, A. Veragua, S. Español-Jiménez, G. Chiang, and F. A. Tobar, "Unsupervised blue whale call detection using multiple time-frequency features," in *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, 2017, pp. 1–6. DOI: 10.1109/CHILECON.2017.8229663.

[20] F. Pace, P. R. White, and O. Adam, "Hidden markov modeling for humpback whale (megaptera novaeanglie) call classification," in *Meetings on Acoustics*, vol. 17, 2012. DOI: 10.1121/1.4772751.

[21] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, *Densely connected convolutional networks*, 2018. DOI: 10.48550/arXiv.1608.06993.

[22] B. S. Miller, S. Madhusudhana, M. G. Aulich, and N. Kelly, "Deep learning algorithm outperforms experienced human observer at detection of blue whale d-calls: A double-observer analysis," *Remote Sensing in Ecology and Conservation*, vol. 9, no. 1, V. Lecours and D. Risch, Eds., pp. 104–116, 2023. DOI: 10.1002/rse2.297.

[23] J. H. Rasmussen and A. Sirovic, "Automatic detection and classification of baleen whale social calls using convolutional neural networks.," *The Journal of the Acoustical Society of America*, vol. 149, no. 5, p. 3635, 2021.

[24] E. Schall, I. I. Kaya, E. Debusschere, P. Devos, and C. Parcerisas, "Deep learning in marine bioacoustics: A benchmark for baleen whale detection," *Remote Sensing in Ecology and Conservation*, vol. 10, no. 5, pp. 642–654, 2024, ISSN: 2056-3485, 2056-3485. DOI: 10.1002/rse2.392.

[25] B. Hoffman and D. Robinson, "Toward in-context bioacoustic sound event detection," DCASE2024 CHallenge, Technical Report, 2024.

[26] P. Zhao, C. Lu, and L.-W. Zou, *Few-shot bioacoustic event detection with frame-level embedding learning system*, 2024. DOI: 10.48550/arXiv.2407.10182.

[27] L. Cances, T. Pellegrini, and P. Guyot, "Multi-task learning and post processing optimization for sound event detection," DCASE2019 Challenge, Technical Report, 2019.

[28] B. Barshan and B. Ayrulu, "Performance comparison of four time-of-flight estimation methods for sonar signals," *Electronics Letters*, vol. 34, no. 16, pp. 1616–1617, 1998. DOI: 10.1049/el:19981127.

[29] J. DiFranco and W. Rubin, *Radar detection* (The Artech radar library). Artech House, 1980, ISBN: 978-0-89006-092-6.

[30] H. Liu, X. Liu, X. Mei, Q. Kong, W. Wang, and M. D. Plumbley, *Surrey system for DCASE 2022 task 5: Few-shot bioacoustic event detection with segment-level metric learning*, 2022. DOI: 10.48550/arXiv.2207.10547.

[31] L. Jean-Labadye, C. Parcerisas, B. Miller, *et al.*, *BioDCASE 2025 task 2: Development set*, version 1 (Erratum), Zenodo, 2025. DOI: 10.5281/zenodo.15092732.

[32] B. S. Miller, K. M. Stafford, I. Van Opzeeland, *et al.*, *An annotated library of underwater acoustic recordings for testing and training automated algorithms for detecting antarctic blue and fin whale sounds*, version 1, Australian Antarctic Data Centre, 2020. DOI: 10.26179/5e6056035c01b.

[33] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems (NIPS 2016)*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Barcelona, Spain: Curran Associates, Inc., 2016.

[34] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 648–656.

[35] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, *Focal loss for dense object detection*, 2018. DOI: 10.48550/arXiv.1708.02002. arXiv: 1708.02002.

[37] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society*, B, vol. 36, no. 2, pp. 111–133, 1974.

Traffic Sign Recognition: Comparing Custom CNNs and Transfer Learning Approaches for Autonomous Vehicle Systems

1st Roland Chuma
Computer Science dept
Rhodes University
Makhanda, South Africa
g20c0027@campus.ru.ac.za

2nd Joshua van Staden
Computer Science dept
Rhodes University
Makhanda, South Africa
joshua.vanstaden@ru.ac.za

3rd Dane Brown
Computer Science dept
Rhodes University
Makhanda, South Africa
d.brown@ru.ac.za

***Abstract*—This study presents a comparative analysis between custom CNN architectures and transfer learning approaches for traffic sign recognition using the German Traffic Sign Recognition Benchmark (GTSRB) dataset. We systematically evaluated performance trade-offs between domain-specific custom CNN designs and pre-trained models, including ResNet-34 and MobileNetV2, across 43 traffic sign classes. The experimental methodology used evaluation metrics including accuracy, precision, recall, F1-score, and inference time analysis. Results demonstrate that while ResNet-34 achieved the highest accuracy at 99.33%, the custom CNN achieved 99.11% accuracy while reducing the inference time by 46.5% compared to ResNet-34 (3.61 vs. 6.75 seconds per iteration). The custom CNN maintained balanced precision (98.99%) and recall (98.88%) metrics, indicating robust performance across traffic sign classes without significant classification bias. These findings challenge conventional preferences for complex pre-trained models in specialised domains, demonstrating that task-specific architectures can achieve optimal performance-efficiency trade-offs. This research provides valuable insights for autonomous vehicle developers, offering empirical evidence that custom CNN architectures represent a more suitable alternative to transfer learning approaches when resources are limited and efficiency is a must.**

***Index Terms*—Object recognition, Neural networks, Transportation**

I. Introduction

Traffic sign recognition (TSR) systems have become a critical component of autonomous vehicle technologies. As the automotive industry moves toward fully autonomous driving, the ability to accurately detect and classify traffic signs in real-time has emerged as a fundamental requirement for road safety and regulatory compliance [1], [2]. The evolution of traffic sign recognition has changed from traditional computer vision approaches using handcrafted features and classical machine learning algorithms to deep learning methodologies, particularly Convolutional Neural Networks (CNNs) [3], [4]. While pre-trained CNN models leveraging transfer learning have shown remarkable success in various computer vision tasks, the specialised nature of traffic sign imagery, which is characterised by distinct geometric patterns and standardised colour schemes, and symbolic representations, suggests potential advantages for custom-designed architectures tailored specifically to this domain [5], [6].

Despite numerous studies applying deep learning to traffic sign recognition, comprehensive studies examining the trade-offs between custom CNN architectures and transfer learning approaches remain limited. Understanding these trade-offs is crucial for practitioners seeking to select optimal approaches based on specific deployment constraints, including computational resources, training data availability, and accuracy requirements. This research addresses this gap by implementing and evaluating a custom CNN architecture on the German Traffic Sign Recognition Benchmark (GTSRB) [7] dataset while conducting performance comparisons, providing insights into inference speed, model complexity, and practical deployment considerations for real-world traffic sign recognition systems.

The rest of the paper is structured as follows: Section II provides an overview of related work in the field of TSR. Followed by Section III, which details the methodology and design, along with the dataset and implementation details. Finally, the results are provided and discussed in Section IV, along with a conclusion in Section V

II. Related Work

A. Traditional Computer Vision Methods

The early development of traffic sign recognition systems relied heavily on traditional computer vision techniques that used handcrafted feature extractors combined with classical machine learning classifiers. These approaches were based on designing feature descriptors that could capture the distinctive characteristics of traffic signs, such as geometric shapes, colour distributions, and textural patterns [8].

Histogram of Oriented Gradients (HOG) emerged as one of the most widely adopted feature extraction methods for traffic sign recognition [9]. The HOG descriptor's ability to capture edge and gradient information made it particularly suitable for detecting the geometric boundaries that define traffic sign shapes. Maldonado-Bascon et al.

[10] demonstrated the effectiveness of HOG features combined with Support Vector Machines (SVMs) for traffic sign classification, achieving reasonable accuracy on early datasets. Similarly, Scale-Invariant Feature Transform (SIFT) descriptors were used to handle variations in scale and orientation of the object, with [11] utilising SIFT keypoints for robust traffic sign detection under varying viewing conditions.

Local Binary Patterns (LBP) provided another avenue for feature extraction, particularly effective for capturing textural information within traffic signs [12]. The computational efficiency of LBP made it viable for real-time applications, leading to its adoption in various TSR systems. Researchers combined multiple feature extraction techniques to create more robust representations, with [13] proposing a multi-feature approach that integrated color, shape, and texture descriptors.

The classification stage in traditional approaches typically employed well-established machine learning algorithms. Support Vector Machines were widely used due to their excellent performance on high-dimensional feature spaces [14]. Random Forest classifiers were also popular, offering the advantage of handling multiple classes naturally and providing feature importance rankings [15]. Some researchers explored template matching approaches, directly comparing extracted features against stored templates of known traffic signs, though these methods were often limited by their sensitivity to variations in lighting, weather, and viewing angles.

B. Deep Learning Techniques

The introduction of deep learning methodologies, particularly Convolutional Neural Networks, marked a revolutionary shift in traffic sign recognition research. Unlike traditional approaches that required explicit feature engineering, CNNs could automatically learn hierarchical feature representations directly from raw pixel data, eliminating the need for handcrafted feature extractors [16].

AlexNet's success on ImageNet [3] demonstrated the superior performance of deep learning over traditional computer vision methods. This breakthrough motivated researchers to explore CNN applications in specialised domains, including traffic sign recognition. Early adopters quickly recognised that the automatic feature learning capability of CNNs was particularly well-suited to the traffic sign domain, where signs exhibit complex combinations of geometric, colour, and symbolic patterns that are difficult to capture with handcrafted features.

Sermanet and LeCun [5] were among the first to apply CNNs to traffic sign recognition, proposing a multi-scale convolutional architecture that could handle variations in sign size and resolution. Their approach demonstrated significant improvements over traditional methods on the GTSRB dataset, achieving state-of-the-art results at the time. The multi-scale approach addressed one of the key challenges in traffic sign recognition, which was the wide variation in sign sizes due to different distances from the camera.

Ciresan et al. [6] further advanced the field with their Multi-Column Deep Neural Network (MCDNN) approach, which combined predictions from multiple CNN columns to achieve even higher accuracy. Their method not only surpassed traditional computer vision approaches but also exceeded human-level performance on the GTSRB dataset. This achievement highlighted the potential of deep learning to solve complex visual recognition tasks that had previously required substantial domain expertise and feature engineering.

The transition to deep learning brought several key advantages over traditional methods. First, the end-to-end learning paradigm eliminated the need for separate feature extraction and classification stages, allowing the entire system to be optimised jointly. Second, CNNs naturally handled translation invariance through weight sharing, making them robust to variations in sign position within images [17]. Third, the hierarchical feature learning enabled CNNs to capture both low-level features (edges, textures) and high-level semantic patterns (shapes, symbols) in a unified framework.

C. Custom CNN Architectures for Traffic Sign Recognition

The development of custom CNN architectures for traffic sign recognition has been driven by the unique characteristics of traffic sign imagery and the specific requirements of automotive applications. Unlike general-purpose image classification tasks, traffic signs have distinct properties that can be exploited through specialised architectural designs. LeNet-based architectures provided the foundation for many early TSR systems [16]. The original LeNet architecture, designed for digit recognition, shared conceptual similarities with traffic sign classification in terms of the structured, symbolic nature of the visual patterns. Researchers adapted LeNet by modifying the number of convolutional layers, filter sizes, and output classes to accommodate the 43-class GTSRB dataset. These adaptations typically involved increasing the network depth and width to handle the greater complexity of traffic signs compared to handwritten digits.

The Multi-Scale CNN approach introduced by Sermanet and LeCun [5] represented a significant architectural innovation specifically designed for traffic sign recognition. Their architecture incorporated multiple parallel pathways processing the input image at different scales, addressing the challenge of recognising traffic signs that appear at various sizes in real-world scenarios. The multi-scale features were combined through a mechanism that allowed the network to leverage both fine-grained details and coarse-grained shape information. This approach achieved remarkable results on GTSRB, establishing a new benchmark for CNN-based TSR systems.

Ciresan et al. [6] used a Multi-Column Deep Neural Network (MCDNN), which used CNN columns with different preprocessing and architectural variations. Each column was designed to capture different aspects of the traffic sign images, such as different contrast enhancements, geometric transformations, or architectural configurations. This approach not only improved accuracy but also provided robustness through diversity, as different columns could compensate for each

other's weaknesses. Their achievement on GTSRB demonstrated the potential of well-designed custom architectures.

Recent developments in custom CNN architectures have incorporated modern deep learning innovations while maintaining focus on the specific challenges of traffic sign recognition. Zhang et al. [18] proposed attention-based CNN architectures that could focus on relevant regions within traffic sign images, improving performance on cluttered or partially occluded signs. The attention mechanisms allowed the network to adaptively weight different spatial locations, enhancing the model's ability to handle challenging scenarios common in real-world driving conditions.

Residual connections, popularised by ResNet [19], have been adapted for traffic sign recognition to enable training of deeper networks without degradation. Custom ResNet variants designed specifically for TSR have incorporated domain-specific modifications such as specialised skip connections that preserve both spatial and colour information crucial for traffic sign classification. These architectures balance the benefits of increased depth with computational efficiency requirements for automotive deployment.

D. Transfer Learning Approaches

Transfer learning has emerged as a powerful alternative to training custom CNN architectures from scratch, leveraging the feature representations learnt from large-scale datasets like ImageNet to improve performance on traffic sign recognition tasks. This approach is particularly valuable when training data is limited or computational resources are limited.

The application of pre-trained models such as VGG (Visual Geometry Group) [4], ResNet [19], and EfficientNet [20] to traffic sign recognition has yielded impressive results with relatively low computational requirements. These architectures, originally trained on ImageNet's diverse image collection, have learnt hierarchical feature representations that can be adapted to the traffic sign domain through fine-tuning strategies.

Transfer learning approaches typically involve replacing the final classification layer of pre-trained models with a new layer suited to the specific problem context [21]. This is followed by training with a reduced learning rate to preserve the valuable low-level features learned on ImageNet while adapting the higher-level representations to traffic signs. Shustanov and Yakimov [22] demonstrated that fine-tuned VGG models could achieve competitive performance on GTSRB with significantly reduced training time compared to training from scratch.

More sophisticated transfer learning strategies have explored layer-wise fine-tuning [23], [24], where different layers of the pre-trained network are updated with different learning rates or frozen entirely. The intuition is that lower layers, which capture universal features like edges and textures, should be preserved, while higher layers need more substantial adaptation to the traffic sign domain. This approach has proven particularly effective when dealing with limited training data or when computational resources are low.



Fig. 1. Sample Images from the GTSRB dataset

Recent work has investigated the transferability of features between different traffic sign datasets, exploring whether models trained on one country's traffic signs can be adapted to another's regulations and sign designs. This cross-domain transfer learning is particularly relevant for developing globally applicable TSR systems that can operate across different geographical regions with varying traffic sign standards.

E. GTSRB Dataset

The GTSRB dataset is the most popular dataset for evaluating traffic sign recognition algorithms, providing a standardised platform for comparing different TSR approaches [7]. GTSRB contains 51,839 images across 43 classes of German traffic signs, representing a diverse collection of prohibitory, danger, and mandatory signs. The dataset exhibits significant variations in image conditions that reflect real-world challenges as shown in Figure 1, including different lighting conditions ranging from bright sunlight to overcast skies, various weather conditions including rain and snow effects, and multiple viewing angles and distances. The image resolution varies from 15×15 to 250×250 pixels, with many images containing signs that occupy only a small portion of the total image area, making the recognition task challenging.

The dataset's class distribution is highly imbalanced, as shown in Figure 2 with some classes containing over 2,000 examples while others have fewer than 200 samples. This imbalance reflects the natural frequency of different traffic signs in real-world scenarios but poses significant challenges for machine learning algorithms that may become biased toward over-represented classes. Speed limit signs, particularly the 50 km/h limit, constitute the largest class, while some warning signs represent the smallest classes in the dataset. Data preprocessing and augmentation techniques have become crucial components of successful TSR systems working with GTSRB. Standard preprocessing includes histogram equalisation to normalise lighting conditions, contrast enhancement to improve the visibility of sign details, and geometric normalisation to standardise sign orientations [2].

Data augmentation strategies have evolved from simple geometric transformations to sophisticated domain-specific techniques. Traditional augmentation includes rotation, trans-

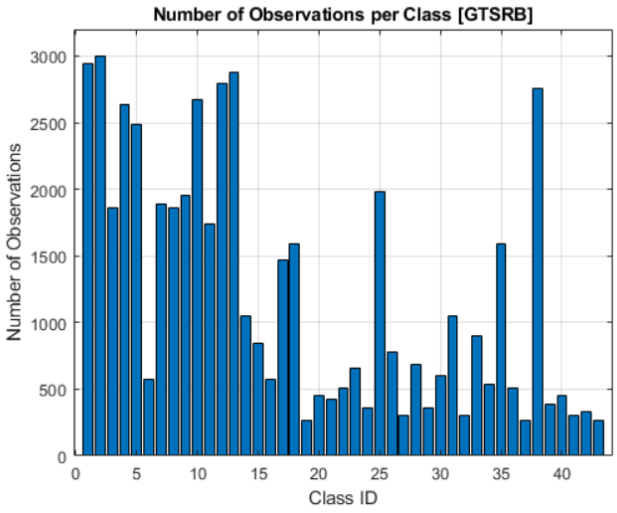


Fig. 2. Frequency Count of Traffic Sign Classes

lation, scaling, and horizontal flipping (where appropriate for sign symmetry). More advanced techniques incorporate realistic distortions such as perspective transformations that simulate different viewing angles, noise injection to simulate sensor limitations, and colour space manipulations to account for different lighting conditions and camera characteristics [25].

The evaluation metrics for GTSRB follow standard multi-class classification metrics, with overall accuracy serving as the primary performance indicator [7]. However, researchers have also examined per-class performance to understand model behaviour across different traffic sign categories. Confusion matrix analysis has revealed common misclassification patterns, such as confusion between similar speed limit signs or difficulty distinguishing between signs with minor geometric differences [26].

State-of-the-art performance on GTSRB has progressed dramatically since the dataset's introduction. Early traditional computer vision approaches achieved accuracies in the 95-98% range, while modern deep learning methods consistently exceed 99% accuracy. The current best-performing systems achieve accuracies above 99.5%, approaching the theoretical limits imposed by label noise and ambiguous images in the dataset.

F. Comparative Studies and Analyses

The literature on traffic sign recognition contains several comparative studies, although most have focused on narrow comparisons between specific approaches rather than comprehensive analyses in different methodological paradigms. These studies provide valuable insights into the relative strengths and weaknesses of different approaches, but often lack the information necessary for practitioners to make better decisions across the wide range of available techniques. Mathias et al. [27] conducted one of the early comprehensive comparisons between traditional computer vision methods and deep learning approaches on GTSRB. Their study compared HOG+SVM, template matching, and early CNN architectures, demonstrating the clear superiority of deep learning methods. However, their comparison was limited to relatively simple CNN architectures and did not explore the full potential of either custom architectures or transfer learning approaches.

More recent comparative studies have begun to address the gap between custom CNN architectures and transfer learning approaches. Dewi et al. [28] compared several pre-trained models (VGG16, VGG19, ResNet50) with custom CNN architectures on GTSRB, finding that while pre-trained models could achieve competitive accuracy with less training time, custom architectures still maintained advantages in terms of final performance and model size. However, their study was limited in scope, comparing only a few representative models from each category. The methodology limitations in existing comparative studies present several challenges for drawing definitive conclusions. Many studies fail to control for important variables such as data preprocessing techniques, augmentation strategies, or hyperparameter optimization procedures. This makes it difficult to attribute performance differences to architectural choices versus implementation details. Additionally, most studies focus solely on accuracy metrics without considering other crucial factors such as computational complexity, training efficiency, or deployment feasibility.

Trade-off analysis has emerged as a critical component in understanding the practical implications of different TSR approaches. Shatnawi et al. [29] examined the accuracy-complexity trade-offs in various CNN architectures, finding that while deeper models generally achieve higher accuracy, the improvement often comes at a significant computational cost that may not be justified for real-time automotive applications. Their analysis highlighted the importance of considering both accuracy and efficiency when selecting appropriate models for practical deployment.

Training efficiency comparisons have revealed significant differences between custom architectures and transfer learning approaches. Custom CNNs typically require extensive hyperparameter tuning and longer training times to achieve optimal performance, while transfer learning approaches can often achieve good results with minimal tuning and shorter training periods. However, custom architectures may ultimately achieve better performance when sufficient computational resources and expertise are available for optimisation. There are multiple gaps in the current comparative literature. Most existing studies compare only a limited subset of available approaches, often within the same methodological family (e.g., comparing different CNN architectures) rather than across paradigms. There is a notable lack of studies that systematically compare custom CNN architectures and transfer learning approaches using identical experimental setups and evaluation criteria.

III. Methodology and Research Design

This research compares custom CNN architectures with transfer learning approaches for traffic sign recognition. Two primary experiments are conducted, firstly an implementation

of a custom CNN architecture designed specifically for traffic sign classification, and secondly, a comparative analysis using pre-trained models adapted through transfer learning on the GTSRB dataset.

A. Dataset Preprocessing

The GTSRB serves as the training and evaluation dataset, containing over 50000 colour images across 43 traffic sign classes [7]. The dataset underwent a series of preprocessing steps designed to improve the generalisability of the model. For training data, the images were first resized at a fixed resolution of 64×64 pixels to ensure uniformity throughout the data set. Data augmentation techniques were then applied to simulate real-world variability and reduce overfitting. Random rotations of up to 15 degrees and random affine transformations with translations of up to 10% were introduced to account for positional and orientation differences in traffic sign appearances. Additionally, colour jittering was used to slightly alter brightness, contrast, saturation, and hue, thereby modelling the varying lighting and environmental conditions encountered in practice. All images were subsequently converted to tensors and normalised using standard ImageNet mean and standard deviation values to stabilise training and speed up convergence. The testing data, in contrast, was only resized, tensorised, and normalised, preserving the original distribution of the dataset to ensure fair evaluation of the model's performance.

B. Custom CNN Architecture

The proposed CNN was developed and trained from scratch, specifically tailored to balance speed and accuracy. To capture fine-grained details like small-font text, the network uses small 3×3 convolutional kernels with unit padding, enabling dense local feature extraction while preserving spatial resolution. The architecture is made up of three convolutional blocks, each progressively increasing the number of filters from 32 to 128, thereby enabling the network to learn low-level edge and colour features in the early layers and more complex shape and texture representations in the deeper layers. Each convolutional layer is followed by batch normalisation and a ReLU activation function, which stabilise training and introduce non-linearity for better feature discrimination. Max-pooling layers are used after each block to reduce spatial dimensions while keeping the most important features, and dropout regularisation is applied to reduce overfitting, an important consideration given the limited number of samples per traffic sign class.

The classifier component begins with an adaptive average pooling layer, which standardises feature map dimensions to 4×4 regardless of input size, enhancing the model's flexibility. The subsequent fully connected layers (512 and 256 neurons) introduce hierarchical abstraction, where learned features are combined to form class-specific representations. Batch normalisation and dropout are again incorporated to improve generalisation and maintain stable gradients. The final linear layer outputs predictions over 43 classes, corresponding to the GTSRB dataset.

C. Transfer Learning Approach

For the transfer learning approach, two experiments were carried out. The pre-trained architectures used are ResNet 34 [30] for residual learning, and MobileNet-V2 (3.4M parameters) for mobile efficiency [31]. Transfer learning follows a two-stage approach. Initially, pre-trained weights are frozen except for the replaced 43-class classification head, using a 0.01 learning rate. During the second stage, fine-tuning unfreezes all layers with a reduced learning rate (0.001) and layer-wise scaling (0.00001 for early layers). Images are pre-processed to 224×224 resolution with ImageNet normalisation for compatibility. The `batch_size` parameter is set to 64 to ensure maximum GPU usage, and the `num_workers` parameter is set to 8 for parallelisation.

D. Performance Metrics

Primary evaluation uses classification accuracy and top-5 accuracy on the test set. Per-class precision, recall, and F1-scores identify systematic biases across sign categories. Confusion matrices reveal misclassification patterns and semantic relationships. Inference time is also a key performance indicator, as this research aims to strike a balance between accuracy and the time taken to classify a road sign.

E. Implementation Environment

All experiments were carried out on Google Colab using Python 3.12 with TensorFlow 2.18 and torch 2.8, maintaining consistent package versions for reproducibility. Deterministic behaviour is ensured through fixed random seeds (Python, NumPy, TensorFlow = 42). Training uses the built-in Colab T4 GPU and 12GB of RAM.

IV. Results and Discussion

A. Experiment 1: custom CNN

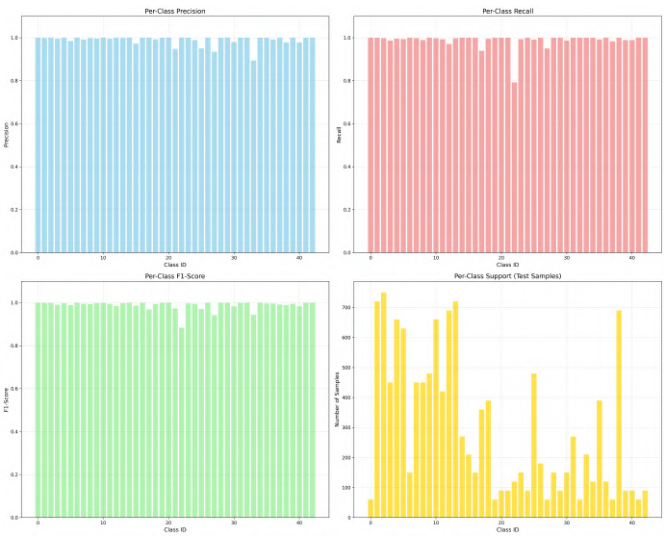


Fig. 3. Per class metrics for the custom CNN, including Precision, Recall and F1 Score.

Figure 3 shows that the CNN model achieved consistently high performance across the 43 traffic sign categories. Precision and recall values are very close to 1.0 for nearly all classes, suggesting that the model is not only making correct predictions but also capturing nearly all true instances of each class. The F1 scores are also high, showing a strong balance between precision and recall, which highlights the model's capabilities to handle various traffic sign patterns despite variations in colour, shape, and size. These results confirm that the data augmentation and regularisation techniques used during training effectively enhanced generalisation.

However, some variation is observed in the support distribution across classes, with certain traffic sign categories being significantly underrepresented in the test set. Despite this imbalance, the model maintained high precision, recall, and F1-scores across both lowly represented and highly represented classes. This demonstrates that the network has learned discriminative features effectively, even for classes with fewer samples. Such consistency across an imbalanced dataset is critical in real-world traffic scenarios, where the frequency of certain signs, like speed limits, greatly outweighs that of others, such as warning or prohibition signs.

B. Experiment 2: ResNet-34

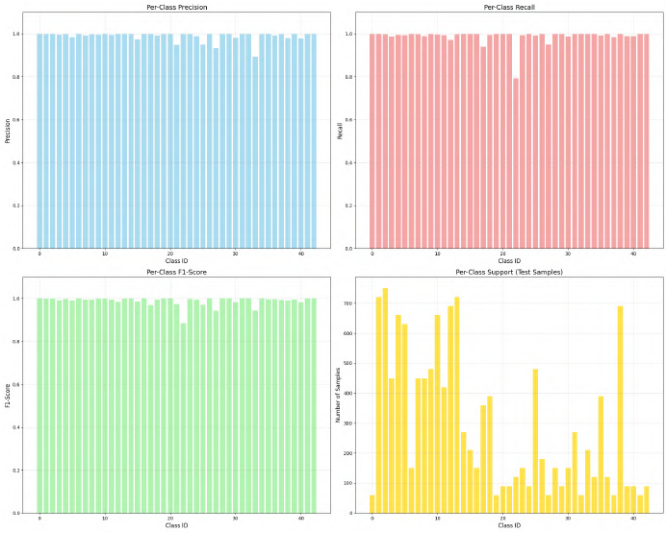


Fig. 4. Per class metrics for ResNet-34, including Precision, Recall and F1 Score.

The ResNet-34 model demonstrates excellent overall performance on the GTSRB dataset, as reflected in Figure 4, which shows consistency across all 43 traffic sign classes. Precision and recall values are largely close to 1.0, suggesting that the model makes very few false positives or false negatives. The high F1-scores further confirm that the model achieves a strong balance between precision and recall, ensuring robust classification performance across diverse traffic sign categories.

The per-class support distribution highlights the inherent class imbalance in the dataset, with some traffic sign classes having significantly fewer samples than others. Despite this imbalance, the model maintains strong performance even on classes with relatively low support, indicating its ability to generalise well without being overly biased toward the more frequent classes.

C. Experiment 3: MobileNet V2

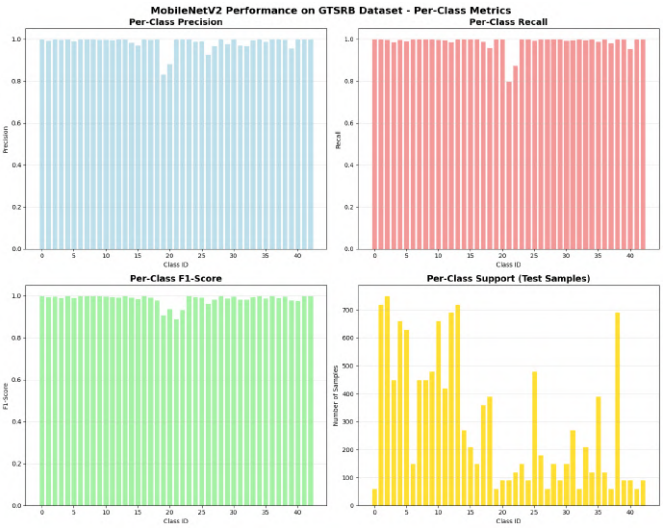


Fig. 5. Per class metrics for MobileNet V2, including Precision, Recall and F1 Score.

The results from MobileNetV2 also show good overall performance, with most classes achieving precision, recall, and F1-scores above 0.9. There are very few classes showing significant performance drops, with only a handful of classes around classes 19 to 21 showing slightly lower recall values around 0.8-0.85. The high precision across nearly all classes indicates that the model has very low false positive rates, making it highly reliable for traffic sign classification applications.

The support distribution reveals the inherent class imbalance in the GTSRB test set, with some classes having over 750 test samples while others have fewer than 100. Interestingly, the model's performance doesn't appear to be strongly correlated with class frequency - even classes with relatively few test samples (like those in the 20-35 range) maintain high precision and recall scores. This suggests that MobileNetV2's architecture and the training process effectively handled the dataset's imbalanced nature, likely benefiting from data augmentation and the model's robust feature extraction capabilities.

D. Performance Comparison across all the models

TABLE I
PERFORMANCE COMPARISON ACROSS ALL THE MODELS

Model	Precision	Recall	F1-Score	Accuracy	Time(s)
Custom CNN	98.99	98.88	98.89	99.11	3.61
ResNet-34	99.35	99.33	99.33	99.33	6.75
MobileNetV2	98.55	98.88	98.60	99.24	1.93

Based on the performance comparison presented in Table I, several significant observations can be made regarding the efficacy of different neural network architectures on the GTSRB dataset.

The results demonstrate that while ResNet-34 achieves the highest overall performance across precision (99.35%), recall (99.33%), F1-score (99.33%), and accuracy (99.33%), the custom CNN presents a compelling trade-off between performance and computational efficiency. The custom CNN achieves competitive accuracy at 99.11%, representing only a 0.22 percentage point decrease compared to ResNet-34, while delivering substantial improvements in inference time. The custom CNN requires only 3.61 seconds to classify a single image compared to ResNet-34's 6.75 seconds, which is a 46.5% reduction in inference time. This computational efficiency advantage becomes particularly important in autonomous driving systems, where rapid decision-making is critical for safety.

When compared to MobileNetV2, which was specifically designed for high-speed computation on mobile and embedded applications, the custom CNN demonstrates balanced performance characteristics. While MobileNetV2 achieves the fastest inference time at 1.93 seconds per iteration, it shows relatively lower precision (98.55%) and F1-score (98.60%) compared to both the custom CNN and ResNet-34. MobileNetV2 uses a highly optimised architecture based on depthwise separable convolutions and inverted residual blocks [31]. This reduces the number of parameters, allowing it to perform feature extraction with fewer floating-point operations and a smaller memory bandwidth, resulting in faster inference times at the cost of lower precision and accuracy.

The custom CNN's performance profile suggests that domain-specific architectural design can achieve an optimal balance between accuracy and speed. Unlike ResNet-34, which employs deep residual connections originally designed for complex natural image classification tasks, the custom CNN appears is more appropriately scaled for the traffic sign recognition domain. The 43-class GTSRB classification task may not require the full complexity of ResNet's 34-layer architecture, allowing the custom CNN to maintain high accuracy while significantly reducing computational overhead.

These results suggest that while pre-trained architectures like ResNet-34 can achieve marginally higher accuracy, custom CNN architectures offer significant advantages in terms of computational efficiency without substantial performance degradation, making them particularly suitable for resource-constrained deployment scenarios in embedded automotive systems.

V. Conclusion and Future Work

This study has presented a comparative analysis between custom CNN architectures and transfer learning approaches for traffic sign recognition using the GTSRB dataset. The experimental results demonstrate that while transfer learning models, particularly ResNet-34, achieve marginally higher accuracy (99.33% vs. 99.11%), custom CNN architectures offer significant computational advantages with 46.5% faster inference times (3.61 vs. 6.75 seconds per iteration). The custom CNN maintains competitive performance across all evaluation metrics, achieving balanced precision (98.99%) and recall (98.88%), while requiring substantially fewer computational resources. These findings suggest that the accuracy gains offered by complex pre-trained models may not justify their computational overhead in resource-constrained environments typical of automotive embedded systems.

The comparative analysis reveals that domain-specific architectural design can effectively balance performance and efficiency requirements for specialised computer vision tasks. Although transfer learning approaches leverage feature representations learnt from large-scale datasets like ImageNet, the relatively constrained domain of traffic sign recognition, characterised by geometric patterns, standardised designs, and controlled visual conditions, appears to be well-suited to lightweight custom architectures. The results indicate that practitioners should consider the deployment context when selecting between these approaches: transfer learning models are preferable when maximum accuracy is paramount and computational resources are abundant, while custom CNNs offer superior value in real-time, embedded applications where inference speed and resource efficiency are critical constraints.

A. Future Work

These are future implementations that will be added to the software artifact.

- Explore optimisation for real-time deployment on edge devices with limited memory and processing capabilities
- Leverage the domain knowledge of custom CNNs and selective transfer learning components to build a model that can generalise across multiple datasets, as road signs vary between countries.

B. Contributions

This study managed to complete the following objective:

- Performance benchmarking: Performed a systematic comparison between custom CNNs and pretrained models, highlighting that domain-specific CNNs offer a more optimal balance between accuracy and efficiency
- Contribution to literature: Provided empirical evidence to aid self-driving industry professionals in making data-driven decisions.
- Optimise the model for use on non-fixed cameras, such as drones and cars
- Make the model detect unusual behaviours or patterns that may help identify crime or dangerous situations

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: a multi-class classification competition," in *The 2011 international joint conference on neural networks*. IEEE, 2011, pp. 1453–1460.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *The 2011 international joint conference on neural networks*. IEEE, 2011, pp. 2809–2813.

[6] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," vol. 32. Elsevier, 2012, pp. 333–338.
[7] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural networks*, vol. 32, pp. 323–332, 2012.
[8] S. Ardianto, C. J. Chen, and H. M. Hang, "Real-time traffic sign recognition using color segmentation and svm," in *2017 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2017, pp. 1–5.
[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 886–893.
[10] S. Maldonado-Bascón, S. Lafuente-Arroyo, P. Gil-Jiménez, H. Gómez-Moreno, and F. López-Ferreras, "Road-sign detection and recognition based on support vector machines," *IEEE transactions on intelligent transportation systems*, vol. 8, no. 2, pp. 264–278, 2007.
[11] G. Loy and N. Barnes, "Fast shape-based road sign detection for a driver assistance system," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1. IEEE, 2004, pp. 70–75.
[12] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
[13] F. Zaklouta and B. Stanciulescu, "Traffic sign classification using k-d trees and random forests," pp. 2151–2155, 2011.
[14] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 1995.
[15] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.
[18] J. Zhang, M. Huang, X. Jin, and X. Li, "Attention-based convolutional neural network for traffic sign recognition," in *2019 Chinese Control Conference (CCC)*. IEEE, 2019, pp. 8620–8625.
[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," pp. 770–778, 2016.
[20] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.
[21] B. Klinton, "Transfer learning: Adapting pre-trained models to new domains," 03 2025.
[22] A. Shustanov and P. Yakimov, "Cnn design for real-time traffic sign recognition," vol. 201. Elsevier, 2017, pp. 718–725.
[23] C. Bu, Y. Liu, M. Huang, J. Shao, S. Ji, W. Luo, and X. Wu, "Layer-wise learning rate optimization for task-dependent fine-tuning of pre-trained models: An evolutionary approach," *ACM Trans. Evol. Learn. Optim.*, vol. 4, no. 4, Nov. 2024. [Online]. Available: https://doi.org/10.1145/3689827
[24] S. Nadipalli, "Layer-wise evolution of representations in fine-tuned transformers: Insights from sparse autoencoders," 2025. [Online]. Available: https://arxiv.org/abs/2502.16722
[25] J. A. Rodríguez-Rodríguez, E. López-Rubio, J. A. Ángel Ruiz, and M. A. Molina-Cabello, "The impact of noise and brightness on object detection methods," *Sensors*, vol. 24, no. 3, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/3/821
[26] L. Gao, Q. Wen, and F. Machida, "Robust and safe traffic sign recognition using n-version with weighted voting," 07 2025.
[27] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, "Traffic sign recognition—how far are we from the solution?" in *The 2013 international joint conference on neural networks (IJCNN)*. IEEE, 2013, pp. 1–8.
[28] C. Dewi, R.-C. Chen, X. Jiang, and H. Yu, "Deep convolutional neural network for enhancing traffic sign recognition dataset," *Neural Computing and Applications*, vol. 32, no. 12, pp. 8685–8699, 2020.
[29] M. Shatnawi, G. Al-Bdour, R. Al-qurran, and M. Al-Ayyoub, "A comparative study of deep learning architectures for traffic sign recognition," in *2020 11th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2020, pp. 196–201.
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385
[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 4510–4520.

CACOphony
Contrastive Architecture for TB Cough classification

Minette Farrell[1], Grant Theron[2], Thomas Niesler[1]
[1]*Department of Electrical and Electronic Engineering*
[2]*DSI-NRF Centre of Excellence for Biomedical Tuberculosis Research and SAMRC Centre for Tuberculosis Research*
[2]*Division of Molecular Biology and Human Genetics, Faculty of Medicine and Health Sciences*
[1,2]*University of Stellenbosch, South Africa*

***Abstract*—We consider the application of contrastive learning to the classification of cough audio recordings to determine a patient's tuberculosis (TB) status. Such classification could form the basis of a low-cost and easily-applied TB triage with which to reduce the incidence of undiagnosed TB, especially in low and middle-income countries with overburdened healthcare systems. We evaluate five non-contrastive baseline classifiers, and apply contrastive training to three of these using both supervised and self-supervised contrastive loss functions. One of these loss functions, exclusive loss, is proposed in this work. We find that improvements over the non-contrastive baselines are possible using several configurations, but that best performance is achieved using a VGGish architecture with additional pre-training and subsequent contrastive training using exclusive loss. This configuration achieves a mean area under the receiver operating characteristic (ROC) of 0.665 on the independent test sets during cross-validation, while the best non-contrastive baseline achieved an AUC of 0.637.**

***Index Terms*—tuberculosis, cough classification, audio classification, contrastive learning, contrastive loss functions, deep learning**

I. Introduction

Tuberculosis (TB) remains the world's deadliest infectious disease, leading to an estimated 1.25 million deaths in 2023 alone [1]. It disproportionally affects low and middle-income countries, including much of Africa. A particular concern is the large number of TB cases that are not detected, and therefore, remain infectious. In response, the World Health Organisation (WHO) has emphasised the need for a triage test that can rapidly and cheaply differentiate between TB positive and TB negative patients [2]. Such a test can provide health care professionals with an immediate result on the basis of which they can decide whether a patient needs further confirmatory TB testing or not. Since the confirmatory testing for TB is costly and requires specialised facilities, an effective triage test can both reduce the burden on healthcare systems and reduce undiagnosed TB cases.

This work concerns the development of such a TB triage test based on the sound of the patient's cough using machine learning. Section II provides some background on the classification of cough audio for TB detection, as well as on the methods that we will employ. Section III describes the datasets we use for experimentation, and Section IV our experimental technique.

Section V presents the experimental results, and Section VI concludes the paper.

We make the following contributions:

- It is the first reported research to use contrastive learning for TB classification using cough audio recordings.
- A novel contrastive loss function, called exclusive loss, is proposed. We show that exclusive loss outperforms existing contrastive loss functions for this application.

II. Background

Medical professionals who were consulted at the start of this study unanimously claimed that they cannot distinguish TB positive from TB negative patients based on the sound of their cough. However, it has been demonstrated that machine learning models can detect aspects of cough audio that allow such distinction to some degree [3]–[8]. Typically, these systems transform the audio signal into a mel-spectrogram or similar time-frequency representation. This serves as input to the machine learning models, which compute a posterior probability indicating whether the patient has TB or not. TB screening using this method has the advantages of being immediate, inexpensive, non-invasive and easy to apply compared to current alternatives such as sputum, skin, blood or X-ray tests.

Contrastive learning can be viewed as a trainable feature extraction procedure which learns to better separate the classes in a transformed feature space before classification is performed. It achieves this by means of contrastive loss functions that explicitly encourage the features to lie close together when the corresponding inputs are similar and further away from each other when they are dissimilar.

A prevalent and successful approach to contrastive learning for image classification is the Simple framework for Contrastive Learning of visual Representations (SimCLR) shown in Figure 1 and described in [9]. SimCLR employs a Siamese network architecture, which consists of two (or more) sub-networks. Each sub-network consists of an encoder that is used to generate representations and a projection head that is used to generate features. Although SimCLR was designed to be trained using NT-Xent loss, it can be adapted to use other contrastive loss functions, such as pair loss, triplet loss and Barlow twins loss. These are introduced briefly below.

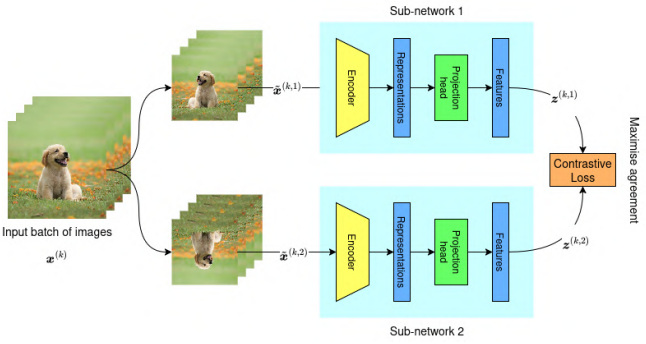


Fig. 1. The SimCLR architecture consists of a Siamese network with two sub-networks, each with an encoder and a projection head. The encoder model is usually a ResNet and is used to generate representations of the input images. The projection head is usually a multi-layer perceptron and is used to generate features from the representations. The contrastive loss function minimises the distances between similar features and maximises the distance between dissimilar features. After the architecture has been trained contrastively, the features are passed to a classifier to generate posterior probabilities.

A. Pair loss

Pair loss is a supervised contrastive loss function that aims to generate features that minimise the distances between similar pairs, while maximising the distances between dissimilar pairs [10].

Let $\boldsymbol{x}^{(k)}$ denote the k_{th} batch of images. The batch is divided into two sub-batches $\tilde{\boldsymbol{x}}^{(k,1)}$ and $\tilde{\boldsymbol{x}}^{(k,2)}$. These image batches are passed to the encoder to generate representations and then to the projection head to generate features $\boldsymbol{z}^{(k,1)}$ and $\boldsymbol{z}^{(k,2)}$. Let $\boldsymbol{z}_i^{(k,1)}$ denote the feature generated by the projection head of the first sub-network for the i_{th} image in the k_{th} batch. Let m denote a margin used to penalise dissimilar distances smaller than this value. Finally, let N_B denote the number of samples in a batch and Y denote an indicator function that uses labels to determine whether a pair is similar (in which case $Y = 1$) or dissimilar ($Y = 0$).

Then pair loss is given by Equation 1.

$$\begin{aligned}\mathcal{L}_{pair_loss}&\left(\boldsymbol{z}^{(k,1)}, \boldsymbol{z}^{(k,2)}; Y\right)\\ &= \sum_{i}^{N_B/2}\Bigg[(1-Y)\times\left\|\boldsymbol{z}_i^{(k,1)}-\boldsymbol{z}_i^{(k,2)}\right\|^2\\ &\quad + Y\times\max\left(0, m-\left\|\boldsymbol{z}_i^{(k,1)}-\boldsymbol{z}_i^{(k,2)}\right\|\right)^2\Bigg]\end{aligned} \tag{1}$$

B. Triplet loss

Triplet loss is a supervised contrastive loss function that extends pair loss by comparing triplets instead of pairs. It aims to learn a feature space that ensures that the distance between similar features is less than the distance between dissimilar features [11].

For every anchor image i drawn from the k_{th} batch, denoted by $\boldsymbol{x}_i^{(k,1)}$, two images are further selected from the same batch. The first has the same label as the anchor and is referred to as the positive image, denoted by $\boldsymbol{x}_i^{(k,2)}$. The second has a different label from the anchor and is referred to as the negative image, denoted by $\boldsymbol{x}_i^{(k,3)}$. Let $\boldsymbol{z}^{(k,1)}$, $\boldsymbol{z}^{(k,2)}$ and $\boldsymbol{z}^{(k,3)}$ denote the batches of features generated by three sub-networks for the anchor, positive and negative images respectively.

Then triplet loss is given by Equation 2.

$$\begin{aligned}\mathcal{L}_{triplet}&\left(\boldsymbol{z}^{(k,1)}, \boldsymbol{z}^{(k,2)}, \boldsymbol{z}^{(k,3)}\right)\\ &= \sum_{i}^{N_B}\max\Bigg(0, \left\|\boldsymbol{z}_i^{(k,1)}-\boldsymbol{z}_i^{(k,2)}\right\|^2\\ &\qquad - \left\|\boldsymbol{z}_i^{(k,1)}-\boldsymbol{z}_i^{(k,3)}\right\|^2 + m\Bigg)\end{aligned} \tag{2}$$

C. NT-Xent loss

Normalised temperature-scaled cross-entropy (NT-Xent) loss is a self-supervised contrastive loss function because it does not require class labels for the training images. Instead, positive samples are derived by applying transformations (referred to as augmentations) to the anchor images. NT-Xent trains the Siamese network to generate features that minimise the distance between augmentations of the same image, while maximising the distance between augmentations of different images [9].

Let $\boldsymbol{z}^{(k,1)}$ and $\boldsymbol{z}^{(k,2)}$ denote the features generated by sub-network 1 and 2 respectively, and let $Z^{(k)}$ denote a feature matrix combining the features in an alternating pattern. Let $\text{sim}(\cdot)$ denote the cosine similarity and let τ denote a temperature parameter.

Then NT-Xent loss is given by Equation 3.

$$\begin{aligned}\mathcal{L}_{NT_Xent}&\left(\boldsymbol{z}^{(k,1)}, \boldsymbol{z}^{(k,2)}\right)\\ &= \frac{1}{2N_B}\sum_{i=1}^{N_B}\left[l(Z_{2i-1}^{(k)}, Z_{2i}^{(k)}) + l(Z_{2i}^{(k)}, Z_{2i-1}^{(k)})\right]\end{aligned} \tag{3}$$

where

$$l\left(Z_{2i-1}^{(k)}, Z_{2i}^{(k)}\right) = -log\left(\frac{e^{\frac{\text{sim}\left(Z_{2i-1}^{(k)}, Z_{2i}^{(k)}\right)}{\tau}}}{\sum_{n=1}^{2N_B}\mathbb{1}_{n\neq\{2i-1\}}\left[e^{\frac{\text{sim}\left(Z_{2i-1}^{(k)}, Z_{n}^{(k)}\right)}{\tau}}\right]}\right)$$

D. Barlow twins loss

Like NT-Xent, Barlow twins is a self-supervised loss because it does not require class labels for the training images. It aims to generate features that are invariant to augmentations and to reduce redundancy between features [12].

Assume again that $\boldsymbol{z}^{(k,1)}$ and $\boldsymbol{z}^{(k,2)}$ denote the features for the two sub-networks. Let α and λ denote hyperparameters that balance the invariance and redundancy terms in the loss and let b denote the b_{th} sample in the batch.

Then Barlow twins loss is given by Equation 4.

$$\begin{aligned}\mathcal{L}_{Barlow_twins}&(\boldsymbol{z}^{(k,1)}, \boldsymbol{z}^{(k,2)})\\ &= \alpha\sum_i(1-C_{ii})^2 + \lambda\sum_i\sum_{j\neq i}C_{ij}^2\end{aligned} \tag{4}$$

where

$$C_{ij} = \frac{\sum_b z_{i,b}^{(k,1)} z_{j,b}^{(k,2)}}{\sqrt{\sum_b (z_{i,b}^{(k,1)})^2} \sqrt{\sum_b (z_{j,b}^{(k,2)})^2}}$$

III. Data and preprocessing

Three independent datasets are used for the development and training of the machine learning models. Nested k-fold cross-validation with $k = 10$ was used for hyperparameter optimisation and evaluation.

Coswara is a publicly available dataset that contains cough audio recordings and diagnoses from COVID-19 positive and negative patients [13]. Using smartphones, cough audio was recorded from 667 COVID-19 positive and 1423 COVID-19 negative patients. We used this data to contrastively pre-train the encoder models.

The **Hyfe** dataset was collected for the CODA TB Dream challenge and contains cough audio recordings and diagnoses from TB positive and negative patients [14]. The cough audio was recorded using the Hyfe application from 297 TB positive and 810 negative patients. We use this data for model development and contrastive pre-training.

Finally, the **CAGE-TB** (Cough Audio triaGE for TB) dataset has been compiled as part of the larger project within which the research reported in this paper falls[1]. This dataset includes cough audio recordings and diagnoses obtained from patients presented at TB clinics around Cape Town, South Africa. Data was recorded using a smartphone and an external USB microphone and comprises 86 positive and 379 negative cases of TB. This dataset was used for both training and evaluation.

Table I shows the number of positive and negative patients and the number of coughs extracted for each dataset.

TABLE I
NUMBER OF PATIENTS AND COUGHS FOR EACH DATASET. COSWARA IS A COVID-19 DATASET, WHEREAS HYFE AND CAGE-TB ARE BOTH TB DATASETS.

Dataset	Samples	Positive	Negative
Coswara	Patients	667	1423
	Coughs	*2789*	*5581*
Hyfe	Patients	297	810
	Coughs	*2930*	*6842*
CAGE-TB	Patients	86	379
	Coughs	*1043*	*4844*

Due to the software used to gather the audio, all cough recordings in the Hyfe dataset are exactly 500ms long. Analysis of the CAGE-TB recordings revealed that it is unusual for coughs to exceed this length. Therefore, recordings for Coswara and CAGE-TB were processed to also be of this length. Recordings shorter than $0.5s$ were padded with trailing zeros, while longer recordings were trimmed.

[1] https://cagetb.org

All recordings were then converted into mel-spectrograms using 2048-sample frames, a hop length of 512 samples and 128 mel-scaled frequency bins, resulting in mel-spectrogram representations with dimensionalities of $[128 \times 43]$.

All augmentations used for contrastive training are applied to these mel-spectrograms. These transformations differ from those commonly used in image processing tasks and are appropriate for audio signals. The augmentations we consider are frequency masking, time masking, cut-out, frequency shifting, time shifting, Gaussian noise addition and solarisation.

IV. Experimental set-up

The classification objective in this work is to determine whether coughs collected from a particular patient indicate TB or not. In order to gauge the degree to which contrastive training is appropriate for this binary classification task, we trained several baseline classifiers directly on the spectrograms.

A. Non-contrastive baseline classifiers

Five classifiers were trained non-contrastively as baselines using cross-entropy loss. The baseline architectures are logistic regression, AlexNet, ResNet18, VGGish and a BiLSTM. Logistic regression is linear model that is included due to its simplicity. AlexNet and ResNet18 are convolutional neural networks that perform well for image classification [15], [16]. VGGish is also a convolutional neural network, but has been adapted from the VGG architecture (which also performs image classification) to the task of audio classification [17]. The BiLSTM is a deep bidirectional recurrent neural network consisting of three bi-directional LSTM layers followed by a dense layer [18]. This architecture is particularly well suited to the modelling of long-term dependencies in time series data. AlexNet and ResNet18 were pre-trained on ImageNet, VGGish was pre-trained on AudioSet, while logistic regression and the BiLSTM were not pre-trained [15], [19].

B. Contrastive architectures

Based on development results, contrastive learning was only implemented for the top three baseline models, namely AlexNet, VGGish and the BiLSTM. Figure 2 shows the architecture used for contrastive learning when using VGGish. A similar configuration was used for AlexNet and the BiLSTM. As shown in the figure, the VGGish feature extractor is used as the encoder and is trained contrastively. The VGGish embedding model is then trained as the classifier using cross-entropy (CE) loss.

C. Contrastive loss functions

All four contrastive loss functions described in Section II were evaluated. In addition, a new contrastive loss function, termed exclusive loss and described below, was also evaluated.

D. Exclusive loss

As described in Section II, pair loss and triplet loss are supervised contrastive loss functions that make use of either a pair of similar inputs or a triplet consisting of a similar and a dissimilar pair.

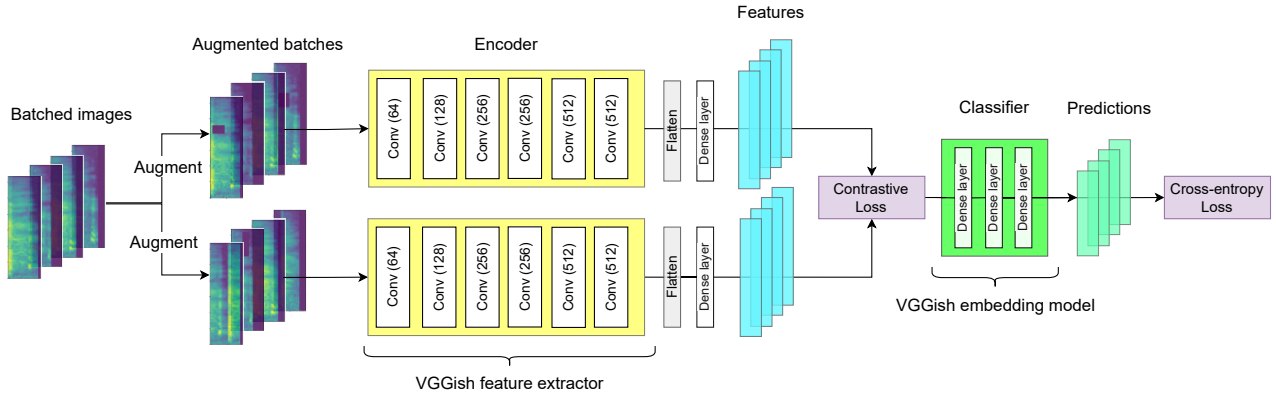


Fig. 2. Contrastive architecture with VGGish, where the VGGish feature extractor is trained using a contrastive loss while the VGGish embedding model is trained with cross-entropy (CE) loss. The mel-spectrogram images are augmented using a random selection of cut-out, solarisation and time shifting.

We propose a new supervised contrastive loss function, exclusive loss, which performs a comparison between every possible sample in a batch, similar to multi-class n-pair and SupCon loss. While other loss functions perform pairwise or triplet-wise comparisons and then aggregate the result, exclusive loss compares all the samples to each other in order to perform a joint minimisation and maximisation.

Let $\boldsymbol{z}^{(k,1)}$ and $\boldsymbol{z}^{(k,2)}$ denote the batches of features for the two sub-networks. Let $\odot$ denote the Hadamand product between two matrices and let $\oplus$ and $\bar{\oplus}$ denote the logical XOR and XNOR operations.

Exclusive loss is then defined in Equation 5.

$$\begin{aligned}\mathcal{L}_{exclusive}&\left(\boldsymbol{z}^{(k,1)},\boldsymbol{z}^{(k,2)},\boldsymbol{y}^{(k,1)},\boldsymbol{y}^{(k,2)}\right)\\ &=\sum\Big[\text{xnor}\left(\boldsymbol{y}^{(k,1)},\boldsymbol{y}^{(k,2)}\right)\odot D\left(\boldsymbol{z}^{(k,1)},\boldsymbol{z}^{(k,2)}\right)\\ &\quad+\text{xor}\Big(\boldsymbol{y}^{(k,1)},\boldsymbol{y}^{(k,2)}\Big)\odot\\ &\qquad\max\left[0,m-D\left(\boldsymbol{z}^{(k,1)},\boldsymbol{z}^{(k,2)}\right)\right]\Big]\end{aligned}\tag{5}$$

where

$$\begin{aligned}\text{xnor}(\boldsymbol{y}^{(k,1)},\boldsymbol{y}^{(k,2)}) &= (\boldsymbol{y}^{(k,1)})\bar{\oplus}(\boldsymbol{y}^{(k,2)})^\top\\ \text{xor}(\boldsymbol{y}^{(k,1)},\boldsymbol{y}^{(k,2)}) &= (\boldsymbol{y}^{(k,1)})\oplus(\boldsymbol{y}^{(k,2)})^\top\end{aligned}$$

In Equation (5), the final loss is calculated by summing the similar elements (XNOR) and the dissimilar elements (XOR) of the Euclidean distance matrix calculated between all features in the batch. The XNOR and XOR operations compare each label to every other label to produce two matrices with the same dimensions as the distance matrix. Computation is efficient since it uses fast bit-wise logical XNOR and XOR operations to select distances for the similar and the dissimilar features respectively.

E. Classifier and training hyperparameters

Nested k-fold cross-validation was used to determine the hyperparameters for the baseline models and contrastive architectures. The Hyfe dataset was used for hyperparameter optimisation, after which the identified hyperparameters were applied to both the Hyfe and the CAGE-TB datasets. AdamW was used as the optimiser, cross-entropy loss was used to train the baseline models and the classifier, while five contrastive loss functions were used to train the contrastive architectures. Table II summarised the hyperparameters chosen based on the development set results.

TABLE II
SET OF HYPERPARAMETERS FOR THE BASELINE MODELS, CONTRASTIVE ARCHITECTURE AND CLASSIFIER BASED ON DEVELOPMENT RESULTS.

Hyperparameter	Baseline	Contrastive encoder	Classifier
N_E	32	64	8
N_B	32	200	100
α	1e-4	1e-5/1e-6	1e-4
β	1e-2	1e-4	1e-2

During model development, the classifier was trained for eight epochs after each contrastive epoch to monitor per-epoch development performance. The contrastive loss functions were trained with $\alpha = 1e-6$, with the exception of pair loss and triplet loss, which was trained with $\alpha = 1e-5$.

V. RESULTS

The contrastive architecture shown in Figure 2 was implemented using the three top-performing baseline models (AlexNet, VGGish and BiLSTM) as encoders based on development set results.

Table III summarises performance in terms of the mean area under the receiver operating characteristic (AUC) calculated over the development folds for CAGE-TB. The first row indicates the baseline results obtained using cross-entropy loss, while the remainder of the results are obtained by training the contrastive architecture in Figure 2 with different contrastive

loss functions. The last two lines show performance when combining a supervised loss function with a self-supervised loss function. We see that the contrastive architecture using VGGish delivered the best performance among the contrastive architectures. Furthermore, exclusive loss delivers the best performance among all the contrastive loss functions for all three architectures.

TABLE III
CAGE-TB DEVELOPMENT SET AUC RESULTS (MEAN OVER ALL DEVELOPMENT FOLDS) FOR CONTRASTIVELY-TRAINED ARCHITECTURES BASED ON ALEXNET, VGGISH AND THE BILSTM.

Loss	AlexNet	VGGish	BiLSTM
Cross-entropy	0.632	0.632	0.625
Pair loss	0.588	0.636	0.609
Triplet loss	0.590	0.635	0.606
Exclusive loss	**0.591**	**0.658**	**0.614**
NT-Xent loss	0.586	0.598	0.610
Barlow twins	0.590	0.608	0.609
Barlow + Triplet	0.586	0.628	0.607
Pair + NT-Xent	0.585	0.612	0.609

VGGish delivers the best baseline and contrastive performance shown in Table III. This VGGish model was the standard implementation pre-trained on AudioSet [20]. A set of experiments was therefore conducted to determine whether further pre-training the encoders with Barlow twins loss can lead to further performance gains. Barlow twins loss was chosen since it is self-supervised and does not require labels, thus other cough datasets can be used instead of TB-only cough datasets. The encoders were further pre-trained on Coswara (COVID-19 dataset) and Hyfe (TB dataset). The best supervised, self-supervised and combination of contrastive loss functions were chosen based on the development results in Table III.

Table IV summarises the mean CAGE-TB AUC results measured over the test folds. The table lists baseline models and contrastive architectures, both with and without further pre-training. We note that the test set performance is within 2.5% of the development set performance seen in Table III.

VGGish delivers the best test set performance among both the baseline models and the contrastive architectures, with and without further pre-training. Furthermore, VGGish trained contrastively with exclusive loss performs best among the considered contrastive loss functions, again both with and without further pre-training. The best overall performance is observed when VGGish is contrastively further pre-trained on Coswara and then fine-tuned contrastively using the novel exclusive loss, leading to a mean test set AUC of 0.665.

Table IV shows that further pre-training leads to improved performance, especially for Coswara but also when using Hyfe. This indicates that the benefit of contrastive training increases as more data is included for training. However, contrastive learning only leads to clear improvements over the baseline when using the VGGish architecture. This indicates that the choice of encoder model is important.

TABLE IV
CAGE-TB TEST SET AUC RESULTS (MEAN OVER ALL TEST FOLDS) FOR CONTRASTIVELY TRAINED ARCHITECTURES BASED ON ALEXNET, VGGISH AND BILSTM, WITH AND WITHOUT FURTHER PRE-TRAINING OF THE ENCODER.

Loss	Standard encoder	Pre-trained on Coswara	Pre-trained on Hyfe
AlexNet			
Cross-entropy	**0.633**	-	-
Exclusive	0.564	0.585	0.600
Barlow twins	0.598	**0.615**	0.608
Barlow+triplet	0.570	0.584	**0.610**
VGGish			
Cross-entropy	0.637	-	-
Exclusive	**0.653**	**0.665**	**0.660**
Barlow twins	0.631	0.617	0.578
Barlow+triplet	0.631	0.648	0.634
BiLSTM			
Cross-entropy	**0.631**	-	-
Exclusive	0.601	0.591	0.596
NT-Xent	0.597	**0.594**	**0.609**
NT-Xent+pair	0.594	0.587	0.594

Finally, the results indicate that further pre-training on cough datasets that are not TB specific might lead to features that are more generalisable and robust.

VI. CONCLUSION

We have considered the application of contrastive learning to the classification of cough audio recordings to determine a patient's TB status. Five baseline models were considered, of which AlexNet, VGGish and the BiLSTM delivered the best performance. Each of these three architectures was then trained contrastively using three supervised, two self-supervised and two combinations of supervised and self-supervised loss functions. Contrastive training of VGGish using the exclusive loss proposed in this paper delivered the best performance. Further pre-training of this configuration using the Coswara and Hyfe datasets led to additional improvements.

We have therefore shown that contrastive learning can be used to compute features that better separate TB positive coughs from TB negative coughs in the feature space. This better separation leads to improved performance of downstream classification of the patient's TB status based on the cough recordings. It was also shown that further pre-training, especially on non-TB cough datasets, can further improve performance. Finally, the new contrastive loss function proposed in this research, exclusive loss, was demonstrated to lead to the largest improvement in classification performance.

VII. ACKNOWLEDGMENTS

We would like to gratefully acknowledge the funding from the EDCTP2 program supported by the European Union (RIA2020I-3305, CAGE-TB) and by Global Health EDCTP3 and its members (101145817, 4-CAGE-TB). We would also like to thank the Telkom CoE for additional support and the Stellenbosch Rhasatsha high performance computing (HPC1) facility for computational resources.

REFERENCES

[1] World Health Organization, "Global tuberculosis report 2024," World Health Organization, Tech. Rep., 2024. [Online]. Available: https://www.who.int/teams/global-tuberculosis-programme/tb-reports/global-tuberculosis-report-2024

[2] World Health Organisation, "High-priority Target Product Profiles for New Tuberculosis Diagnostics: Report of a Consensus Meeting," Report, 2014.

[3] G. Botha, G. Theron, R. Warren, M. Klopper, K. Dheda, P. Van Helden, and T. Niesler, "Detection of tuberculosis by automatic cough sound analysis," *Physiological measurement*, vol. 39, no. 4, p. 045005, 2018.

[4] M. Pahar, M. Klopper, B. Reeve, R. Warren, G. Theron, and T. Niesler, "Automatic cough classification for tuberculosis screening in a real-world environment," *Physiological measurement*, vol. 42, no. 10, p. 105014, 2021.

[5] M. Pahar, M. Klopper, B. Reeve, R. Warren, G. Theron, A. Diacon, and T. Niesler, "Automatic Tuberculosis and COVID-19 Cough Classification Using Deep Learning," in *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*. Istanbul, Turkey: IEEE, 2022, pp. 1–5.

[6] G. D. Yellapu, G. Rudraraju, N. R. Sripada, B. Mamidgi, C. Jalukuru, P. Firmal, V. Yechuri, S. Varanasi, V. S. Peddireddi, D. M. Bhimarasetty *et al.*, "Development and clinical validation of Swaasa AI platform for screening and prioritization of pulmonary TB," *Scientific Reports*, vol. 13, no. 1, p. 4740, 2023.

[7] G. T. Frost, G. Theron, and T. Niesler, "TB or not TB? Acoustic cough analysis for tuberculosis classification," in *Proc. Interspeech 2022*, 2022, pp. 2448–2452.

[8] G. P. Kafentzis, S. Tetsing, J. Brew, L. Jover, M. Galvosas, C. Chaccour, and P. M. Small, "Predicting tuberculosis from real-world cough audio recordings and metadata," *arXiv preprint arXiv:2307.04842*, 2023.

[9] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PmLR, 2020, pp. 1597–1607.

[10] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality Reduction by Learning an Invariant Mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, New York, USA, 2006, pp. 1735–1742.

[11] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[12] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *International conference on machine learning*. PMLR, 2021, pp. 12 310–12 320.

[13] D. Bhattacharya, N. K. Sharma, D. Dutta, S. R. Chetupalli, P. Mote, S. Ganapathy, C. Chandrakiran, S. Nori, K. Suhail, S. Gonuguntla *et al.*, "Coswara: A respiratory sounds and symptoms dataset for remote screening of SARS-CoV-2 infection," *Scientific data*, vol. 10, no. 1, p. 397, 2023.

[14] D. Jaganath, S. K. Sieberts, M. Raberahona, S. Huddart, L. Omberg, R. A. Rakotoarivelo, I. N. Lyimo, O. Lweno, D. J. Christopher, N. Viet Nhung *et al.*, "Accelerating cough-based algorithms for pulmonary tuberculosis screening: Results from the coda tb dream challenge," *medRxiv*, pp. 2024–05, 2024.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25 (NIPS)*. Lake Tahoe, NV, USA: Curran Associates, Inc., 2012, pp. 1097–1105.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "CNN architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2017, pp. 131–135.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[20] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[21] World Health Organization, "Tuberculosis," 2023. [Online]. Available: https://www.who.int/health-topics/tuberculosis

[22] K. K. Lella and A. Pja, "Automatic diagnosis of COVID-19 disease using deep convolutional neural network with multi-feature channel from respiratory sound data: cough, voice, and breath," *Alexandria Engineering Journal*, vol. 61, no. 2, pp. 1319–1334, 2022.

[23] N. K. Sharma, S. R. Chetupalli, D. Bhattacharya, D. Dutta, P. Mote, and S. Ganapathy, "The Second DiCOVA Challenge: Dataset and performance analysis for COVID-19 diagnosis using acoustics," *arXiv preprint arXiv:2110.01177*, 2021.

[24] N. K. Sharma, A. Muguli, P. Krishnan, R. Kumar, S. R. Chetupalli, and S. Ganapathy, "Towards sound based testing of COVID-19—Summary of the first Diagnostics of COVID-19 using Acoustics (DiCOVA) Challenge," *Computer Speech & Language*, vol. 73, p. 101320, 2022.

[25] L. Kranthi Kumar and P. Alphonse, "COVID-19: respiratory disease diagnosis with regularized deep convolutional neural network using human respiratory sounds," *The European Physical Journal Special Topics*, vol. 231, no. 18, pp. 3673–3696, 2022.

[26] ——, "COVID-19 disease diagnosis with light-weight CNN using modified MFCC and enhanced GFCC from human respiratory sounds," *The European Physical Journal Special Topics*, vol. 231, no. 18, pp. 3329–3346, 2022.

[27] M. Pahar, M. Klopper, R. Warren, and T. Niesler, "COVID-19 Cough Classification Using Machine Learning and Global Smartphone Recordings," *Computers in Biology and Medicine*, vol. 135, p. 104572, 2021.

[28] ——, "COVID-19 Detection in Cough, Breath and Speech Using Deep Transfer Learning and Bottleneck Features," *Computers in Biology and Medicine*, vol. 141, p. 105153, 2022.

[29] M. Pahar, I. Miranda, A. Diacon, and T. Niesler, "Automatic Non-Invasive Cough Detection Based on Accelerometer and Audio Signals," *Journal of Signal Processing Systems*, vol. 94, no. 8, pp. 821–835, 2022.

[30] J. Anton, H. Coppock, P. Shukla, and B. W. Schuller, "Audio barlow twins: Self-supervised audio representation learning," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

Core Network Technologies and Architecture

Deep Learning vs Traditional Learning for Ransomware Detection: A Performance Comparison Analysis

1st Emmanuel Musiiwa
Computer Science Department
Rhodes University
Makhanda, South Africa
g20m4048@campus.ru.ac.za

2nd Stones Dalitso Chindipha
Computer Science Department
Rhodes University
Makhanda, South Africa
s.chindipha@ru.ac.za

3rd Moses Moyo
Information Systems Department
Rhodes University
Makhanda, South Africa
moses.moyo@ru.ac.za

***Abstract*—Ransomware poses a threat to organisations and individuals as it can cause disruptions to operations and permanent data loss necessitating robust detection methods. This study addressed this need by performing a comparative performance analysis between deep learning and traditional learning models for ransomware detection. The study utilised two datasets, EldeRAN and MLRan, to evaluate five models: Random Forests, SVM (with linear kernel), Logistic Regression, Multi-layer perceptron and TabNet. The performance metrics used to evaluate the models were accuracy, F1-score, precision and recall. The findings of this study show that Logistic regression had the best performance with 0.9973 accuracy, F1-score, precision and recall of 0.9971. The second best model was MLP with 0.9959 accuracy, F1-score of 0.9957, precision of 0.9943 and recall of 0.9971. It was observed that traditional learning models remained sturdy and had minimal sensitivity to different feature sizes. Random Forests' accuracy of 0.9869, F1-score of 0.9829, precision of 0.9773 and recall of 0.9885 using 483, 800 and 1,200 features, showing how feature dimensionality does not affect performance. Deep learning models, however, showed mixed sensitivity with the Multi-layer perceptron, achieving improved performance with an increase in the number of features, highlighting their potential to take advantage of additional information while the performance of TabNet plummeted as the number of features increased. With 483 features, MLP had an accuracy of 0.9738, F1-score, precision and recall of 0.9655. These metrics improved to 0.9869 for accuracy, 0.9827 for F1-score, 0.9884 for precision and 0.9770 for recall using 1,200 features. The study concluded that both traditional learning and deep learning models perform well for ransomware detection, but their applicability depends on the trade-off that is required.**

***Index Terms*—Ransomware detection, traditional learning, deep learning, machine learning, random forests, logistic regression, SVM, multi-layer perceptron, TabNet**

I. Introduction

Ransomware, a subtype of malware, encrypts user files on a system, rendering them inaccessible unless a ransom for the decryption key is paid [1], [2]. Due to its ability to lock important data, ransomware poses a threat to organisations, individuals and the government as it can cause disruptions to operations and permanent data loss [3]. With the workplace paradigm shifting to a more remote system, the rise of ransomware attacks has exposed the vulnerabilities that exist in organisational, and government infrastructure but as well as highlight the limitations of existing detection measures [4], [5]. Furthermore, the use of Ransomware as a Service (RaaS) has lowered the barriers to entry by enabling less technically adept attackers to make use of ransomware without creating it [6], [7]. To use RaaS, an attacker pays a fee to the ransomware author to deploy the variant for an attack [8], [9].

Different detection methods, including traditional (static) detection, behavioural detection and use of machine learning have been proposed to combat ransomware [10]. Traditional detection methods rely on unique patterns, signatures, in ransomware code for detection [4]. Traditional detection systems are effective against known ransomware variants but struggle against novel variants as attackers continuously refine their methods [3], [11]. If the signatures are not updated, newer ransomware variants may not be detected [3]. This limitation necessitates a more adaptive approach that does not depend on predefined signatures and patterns for detection due to the ever-changing nature of the ransomware landscape [4], [10].

Behavioural detection involves analysis of ransomware during execution to derive behaviour that indicates malicious activity on the host machine [4], [10]. Sandbox environments are used to analyse ransomware and extract features that are indicative of anomalous behaviour [12]. This method bypasses the patterns that are used for static detection as it focuses on how ransomware behaves during execution [10]. Although being more advantageous than static detection, behavioural analysis still has its limitations. When faced with goodware that mimics ransomware-like behaviour, misclassification occurs, which leads to false positives and false alarms may be triggered, disrupting normal operations [4], [5]. Additionally, real-time monitoring of a machine to distinguish between anomalous and normal behaviour may be resource-intensive.

Machine learning for detection has shown promise in the ransomware detection field. Machine learning models are able to find patterns in the data that help to distinguish between goodware and ransomware [10]. Using machine learning for detection has shifted from reactive detection to proactive detection approaches [5], [8], [10]. To train machine learning models, extensive datasets are required [10]. Due to the ability of models learning continuously, machine learning becomes

a vital component for creating robust detection tools.

Recent advances in ransomware detection field have led to the development of hybrid models that combine behavioural analysis and machine learning to overcome the limitations of the two approaches individually while utilising the strengths that each method offers [3]. This study investigates which machine learning models (deep learning vs traditional learning) are effective for the detection of ransomware using dynamic features. To achieve this feat, a performance comparison of accuracy, precision, recall, and F1-score between representative traditional and deep learning algorithms will be done. The objectives of this study are:

1) Compare the performance of deep learning and traditional machine learning models for ransomware detection
2) Determine the influence of feature selection techniques on the performance of these models

The rest of this study is organised as follows: Section II discusses machine learning architectures (traditional and deep learning models) that have been used for ransomware detection using dynamic indicators, outlining their similarities and differences. Section III outlines the methodology, including the data collection and pre-processing, model selection and implementation, the performance metrics used and the experimental environment on which the study was done. Section IV presents the findings obtained from the experiments, with Section V discussing and interpreting these findings. Finally, Section VI concludes the paper with a summary of the findings and suggesting areas for future work.

II. Machine Learning Architectures

Using machine learning for ransomware detection has shown that this approach has the ability to identify novel ransomware variants through recognising complex patterns within data [3], [13], [14]. This section outlines the definitions and characteristics that distinguish deep learning from traditional learning.

A. Traditional Learning

Traditional learning (TL) models refer to machine learning models that have limited architectural depth, usually with one or two processing layers [13], [15]. The purpose of such models is to identify the patterns and relations that are in the input data to produce output or predictions. As a result, these models have lesser parameters to learn, which leads to their computational efficiency and the ease of hyperparameter tuning [13]. Another key feature of traditional learning models is their reliance on manual feature engineering [14], [16]. Feature engineering is a process where useful features are extracted from raw data before a model can be trained, a process which is done by domain experts [1], [4], [14], [16], [17]. Traditional learning models tend to be used on smaller datasets as they are more efficient when data is limited [14]–[16], unlike deep learning models, where overfitting is an issue. Adding on, the simplicity that comes with traditional learning models often means better interpretability of results, giving clear insights into their decision-making process [8], [15], [18]. This simplicity also leads to lower use of computational resources, which leads to quick and cost-effective training.

Some traditional learning models that have been used for ransomware detection include Random Forests, Support Vector Machines (SVM), and Logistic Regression [19]. Random forest models have been used to detect ransomware through the analysis of behavioural indicators such as the frequency of packets and their size in an encrypted file-sharing network [13]. Similarly, [20] used random forests to effectively capture complex interactions between features indicative of ransomware. [21] used Bayesian networks to model probabilistic relationships between system events, which enabled the detection of ransomware with notable precision. Clustering algorithms have also been used to group similar ransomware behaviours, aiding in the identification of new ransomware variants through anomaly detection [22].

Despite bringing about a change in the ransomware detection landscape, traditional machine learning has some drawbacks. The reliance on manual feature engineering requires a domain expert and can be time-consuming. Feature engineering can also limit the ability to generalise on unseen data. Adding on, when data is complex and large, these models struggle to generalise and scale on large datasets.

B. Deep Learning

Deep learning (DL) models refer to machine learning models that make use of several layers of neural networks to perform tasks such as classification and regression [3]. These models are termed deep due to the use of multiple layers, which span from three to several hundred within the same architecture. Using deep learning, models are constructed through mimicking the structure of the human brain [5], [18], [21]. Deep learning offers the advantage of automatic feature extraction from data, which is often referred to as representation learning [18]. Through representation learning, deep learning models can learn which features are relevant to produce output (classification, regression) which eliminates manual feature engineering. This, in turn, reduces the time that is required to do data preparation, a labour-intensive process. Deep learning models have capabilities of performing well on tasks that require discovering underlying patterns and hidden relationships within data [5], [21], [23]. Their use has gone beyond that of numerical data analysis, enabling effective processing of image, speech and other forms of unstructured data. Adding on, deep learning models are data hungry and tend to perform better as more data is provided. This makes them ideal for use with large scale datasets and applications.

Some deep learning models that have been used for ransomware detection include Long-Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs), among others. [5], [24] focused on the use of LTSM (Long Short-Term Memory) networks and file entropy analysis to detect ransomware. LSTM networks provide sequential learning capabilities [18], and entropy-based analysis

was used to capture the distinct behavioural patterns that emerge during the execution of ransomware [5]. Similarly, [18] made use of CNNs to detect ransomware using I/O operations (short Control Commands, bytes being read, and written) per second over a period of time. [21] used Generative Adversiral Networks (GANs) to analyse I/O Request Packet (IRP)operations. This study highlighted the use of GANs as a powerful tool in the ransomware detection landscape.

Although offering several advantages, deep learning also comes with significant limitations. Deep learning models are characterised by high computational costs as they are highly resource-intensive [5], [18], [21]. Training deep learning models also often requires high-performance hardware such as Graphical Processing Units (GPU) or Tensor Processing Units (TPU) to efficiently process data. Adding on, deep learning models are less interpretable which leads to difficulties in explaining how a result was obtained [21]. This lack of interpretability can be an issue in applications which require understanding a model's decisions. Another issue is overfitting, which occurs when deep learning models are used on small datasets.

C. Differences between deep and traditional learning

The main differentiating factor between deep learning models and traditional learning models is the number of hidden layers within each architecture. Traditional learning models make use of one or two layers, while deep learning models make use of several layers ranging from three to several hundred [5], [8], [15], [18]. Another factor used to differentiate between traditional and deep learning is the approach to feature engineering [14], [18], [21]. When using traditional learning, manual feature engineering is done, which requires a domain expert to extract relevant features from raw data. On the other hand, deep learning has an automated feature selection process, learning relevant features from raw data by itself without input from a domain expert in doing so.

Deep learning is effective with larger datasets, which are required to produce optimal performance and avoid overfitting [18]. Traditional learning, on the other hand, performs well on small datasets and is less prone to overfitting in such cases [1], [4]. With regards to computational resource usage, deep learning demands more resources due to its complex architectures and numerous parameters, often requiring special hardware like GPUs. Flipside, traditional learning is less resource-intensive and typically takes less time to train.

While both perspectives of machine learning have been applied within the ransomware detection landscape, a direct comparative analysis, which explicitly weighs the trade-offs between the two, is valuable.

A summary of the differences between traditional and deep learning is presented in Table I.

TABLE I
COMPARISON BETWEEN DL AND TL

Aspect	TL	DL
Architectural Depth	1–2 layers	3 to hundreds
Feature Engineering	Manual	Automatic
Interpretability	High	Low
Resources	Low – efficient	High – GPU/TPU intensive

III. METHODOLOGY

This section outlines the experimental design, dataset(s) used, selected models and the metrics used for evaluation. Fig 1 shows the methodology outline which is designed to provide a comprehensive comparison between deep learning and traditional learning models for ransomware detection. The approach starts off with the collection of publicly available dynamic datasets for ransomware detection. Data preparation then follows, with model training happening after. During model training, model optimisation is done. An outline of the optimisation is shown in Fig 2. The process starts off with taking the data and then performing feature selection before splitting the data. The training data is used to build the models, while the validation set is used for hyperparameter tuning and finally evaluation is done using the test set.

A. Datasets

This study used two publicly accessible datasets, EldeRAN[1] and MLRan[2]. These datasets are made up of features which were collected through dynamic analysis of ransomware.

1) EldeRAN dataset: EldeRAN was proposed by [16] and published in 2016. The dataset has 582 ransomware variants across 11 families and 942 samples of legitimate software. Both ransomware and goodware were run in isolation using Cuckoo Sandbox with a Windows XP 32-bit operating system. The execution of ransomware and goodware led to 30,967 features (which include API system calls, file extensions, embedded strings, directory activities, file system, dropped files, and registry operations) being extracted. EldeRAN has an imbalance as it uses more goodware samples as compared to ransomware. The selected features are represented by a binary value (1 or 0) indicating the occurrence of a certain system event.

For the EldeRAN dataset, feature selection was done using ANOVA F-values to use only features that were informative for model training. The SelectKBest method was used to put features in order of their significance in distinguishing ransomware from legitimate software to evaluate how feature selection affetcs model performance. The number of features selected was 483, 800 and 1,200.

2) MLRan dataset: [14] proposed MLRan, a ransomware dataset compiled using dynamic features. This dataset was made up of 4,880 samples of both ransomware and goodware. Ransomware samples collected span from 2006 to 2024, making up 2,330 samples from 64 families. Goodware made

[1]https://github.com/rissgrouphub/ransomwaredataset2016
[2]https://github.com/faithfulco/mlran

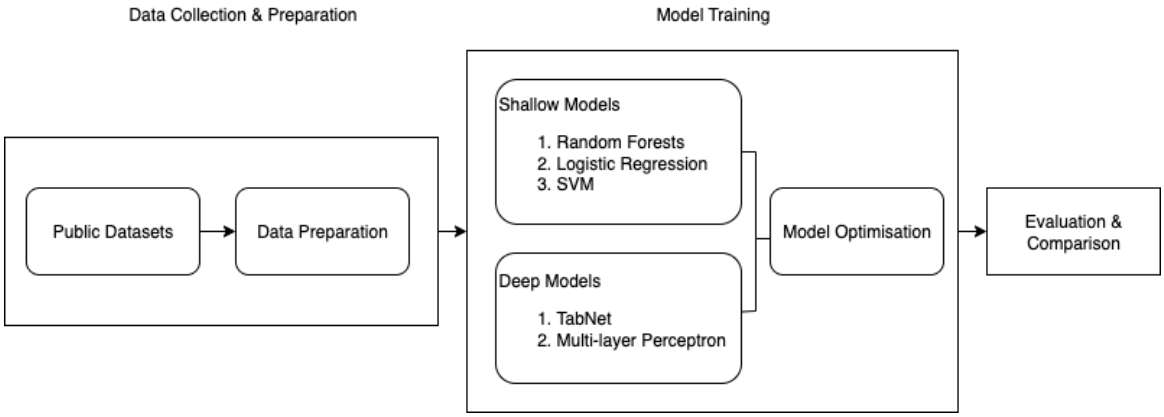


Fig. 1. An overview of the methodology.

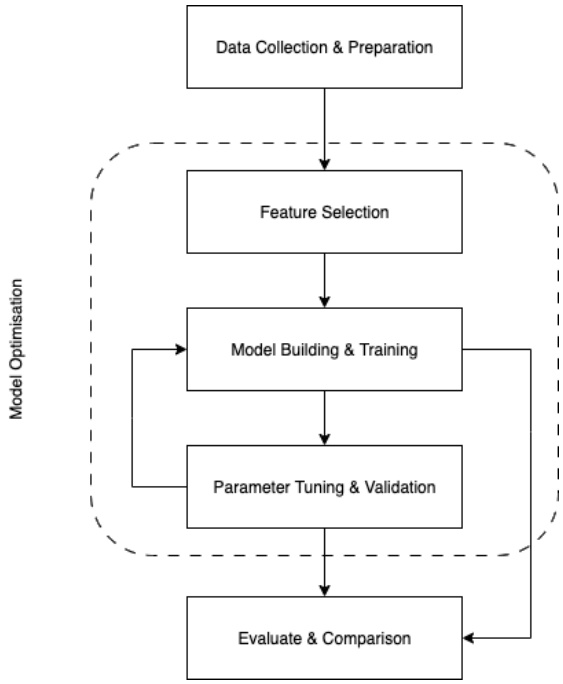


Fig. 2. An overview of the model optimisation.

up 2,550 samples. Dynamic analysis of samples was done on Cuckoo Sandbox with a Windows 7 operating system on top, to generate detailed reports of sample behaviour during execution. From these reports, API calls, registry key operations, file operations, directory operations, extracted strings, network activity, system resource usage, dropped file extensions and types, and dynamic signatures were the distinct feature groups that were extracted.

These two datasets were selected due to the vast number of features that they have. During data preparation, non-predictive columns such as “ID”, “sample type”, “family type”, and the “Ransomware Family” column were kept for analysis but were not included as features for the binary classification task. All features that were all 0 across all samples were dropped. These features were discarded because they did not carry any information that could be used for the binary classification task. The datasets were partitioned using the 70:15:15 split for training, validation and testing, respectively. This split was used so that the validation set can be used to tune hyperparameters and improve model performance. To ensure that the number of samples of ransomware and goodware in the training, validation and testing sets was maintained, stratified sampling was used.

B. Model selection and implementation

To conduct a performance comparison, a selection of three traditional machine learning and two deep learning models was made. Traditional models were selected because they are widely used and are regarded as strong baselines in tabular data and have been used in other literature [14], [16], [25]. Deep learning models were selected to evaluate more complex multi-layered architectures.

All deep learning and traditional learning models were implemented in Python. To handle the class imbalance posed by the EldeRAN dataset, the class_weight parameter was set to ”balanced” within the scikit-learn library for the traditional models. This adjustment gives more weight to the minority (ransomware) class during training. For the deep learning models, sample-level reweighting was used to make the models pay more attention to the minority class (ransomware).

1) Traditional learning models: Logistic regression (LR), Random forests (RF) and Support Vector Machines (SVM) were chosen as the traditional machine learning models. These are high-performing algorithms on tabular data, and their implementation is relatively simple.

- Logistic regression - a linear model used for binary classification tasks. It offers high interpretability and assumes that a linear relationship exists between features and log-odds of an outcome. In this study, L2 regularisation was used to prevent overfitting on high-dimensional feature sets. The EldeRAN dataset posed a class imbalance between ransomware and goodware which was addressed through setting the class weight to “balanced” to counteract class imbalance by giving more weight to the minority class. To implement the model, the scikit-learn library was used.

- Support vector machine - works through finding a hyperplane to separate data points. The SVM used in this study is linear to cater for complex relations in the datasets, computational efficiency and high performance in high-dimensional spaces. In this study, the class imbalance posed by the EldeRAN dataset was handled through setting the class weight to balanced to ensure the model did not favor the majority class . The Scikit-learn library was used to implement the model.
- Random Forests - RF makes use of multiple decision trees to produce an output. They are highly effective with data that has non-linear feature relations. The RF model used in this study consists of 100 decision trees to ensure a stable and accurate ensemble prediction and was configured with no maximum depth to allow for comprehensive learning. Similar to SVM and LR, the class weight was set to balanced to account for class imbalance, ensuring the minority class was fairly represented in each tree's decision process. The model was implemented using the scikit-learn library.

2) Deep learning models:

- Multi-layer perceptron (MLP) - it is a feedforward neural network. In this study, the model consists of an input layer, two hidden layers of 100 and 50 neurons respectively and the output layer. A Rectified Linear Unit (ReLU) activation function was applied to the hidden layers to allow MLP to learn and approximate the relationships within the data. Sample weighting was used to address class imbalance by forcing the model to focus more on the minority class, and scikit-learn was used to implement the model.
- TabNet - a deep learning model specifically made for tabular data [26]. It makes use of sequential attention to choose the features to reason from at every processing step. In this study, the model was configured with 3-5 steps, 8-16 dimensions and an entmax masking function to make the model's decision-making more efficient and interpretable. To handle class imbalance, sample-level reweighting was used to counteract imbalance during the training process. The PyTorch TabNet library was used to implement the model.

C. Performance metrics

To get an extensive evaluation of the models used in this study, accuracy, precision, recall and F1-score were used. These metrics provide a way to assess the effectiveness of the model, which is crucial in the ransomware detection landscape where the penalty of misclassification (causing a false alarm or not detecting a threat) can be severe. The following metrics were selected for model evaluation:

- Accuracy - measures the number of correct predictions made by a model.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- Precision - measures the number of correctly identified positives (True Positives) to the total number of instances predicted as positive (True Positives + False Positives). High precision means a reduced number of false positives (false alarms).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

- Recall - measures how many of the actual positive cases were identified by a model. It compares the actual positive cases (True Positives) to the total number of actual positives (True Positives + False Positives). A high recall score means less threats are missed.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

- F1-score - measures the trade-off between recall and precision. F1-score is important when using imbalanced datasets due to its penalising models that favour one metric over another. High F1-score means a model is good at predicting and at detecting all threats.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

D. Experimental environment

The models selected in this study were all trained and evaluated on the same system. The computer hardware specifications are as follows:

- CPU: 12th Gen Intel(R) Core(TM) i5-12400 (2.50 GHz)
- RAM: 16GB (15,7 GB usable)

IV. Results and Findings

This section provides the quantitative analysis of the performance of the machine learning models used for this study. The evaluation of the models on the EldeRAN dataset includes multiple feature size variants (483 features, 800 features and 1,200 features) to determine whether the number of features is a contributing factor to the performance of the model. The quantitative performance comparison examines the metrics mentioned in Section III-C.

A. Model Performance on MLRan

The machine learning models were trained separately on the MLRan dataset to determine how they perform in ransomware detection. Table II shows the results obtained by each model on the MLRan dataset

TABLE II
Performance results on MLRan

Model	Accuracy	F1-Score	Precision	Recall
LR	**0.9973**	**0.9971**	**0.9971**	**0.9971**
SVM	0.9918	0.9915	0.9886	0.9943
RF	0.9863	0.9858	0.9802	0.9914
MLP	0.9959	0.9957	0.9943	**0.9971**
TabNet	0.9904	0.9900	0.9886	0.9914

On the MLRan dataset, all the models achieved great scores for the binary classification task. With the F1-score being lowest at 0.9886, this indicates that the dynamic features that were used in this dataset have a strong signal, which is effective for distinguishing between ransomware and goodware.

Results in Table II show that LR model had the best performance with an F1-score of 0.9971, coupled by the best precision and recall, balanced at 0.9971. Accuracy stands at 0.9973 for LR, making it the standout model as it outperformed all the other models. These results highlight that despite the simplicity of LR, it is capable of capturing crucial relationships within the dataset, making it good for identifying unusual behaviour caused by ransomware in a system from normal system behaviour from goodware.

MLP trailed LR in terms of F1-score at 0.9957, with precision and recall at 0.9943 and 0.9971, respectively. These metrics indicate that it is highly effective at distinguishing between ransomware and goodware. A high recall of 0.9943 is an important score, as missing a ransomware attack can lead to severe costs. With 0.9943 precision, the model slightly lags behind LR, which may hint at an increase in false positives, but, is still highly competitive.

TabNet and SVM showcase almost similar performance metrics, but lower than those of MLP and LR. SVM achieved a slightly higher F1-score of 0.9915, with TabNet achieving 0.9900. The recall score for SVM was 0.9943, with TabNet scoring 0.9914. The precision of the two model was 0.9986 with accuracy at 0.9918 for SVM and 0.9904 for TabNet. From Table II, it can be noted that both models achieved a recall score higher than precision, which implies that the models may prioritise identifying ransomware at the expense of increased false positives. Such a trade-off may be tolerated in contexts where missing ransomware may lead to more costs as compared to classifying legitimate software as ransomware.

Although tied up with TabNet for recall at 0.9914, RF had the lowest performance metrics amongst the five models. It achieved an accuracy score of 0.9863, F1-score of 0.9858 and precision of 0.9802. With a precision score that is lower than the recall, this suggests that the model may be very sensitive to ransomware behaviour, detecting almost all the ransomware but at a cost of increased false positives.

Fig 3 illustrates a visual performance of the results presented in Table II on the MLRAN dataset. LR outperforms all the models for accuracy, recall, F1-score and precision. MLP trails LR slightly in terms of accuracy, F1-score, and precision, but had the same score for recall.

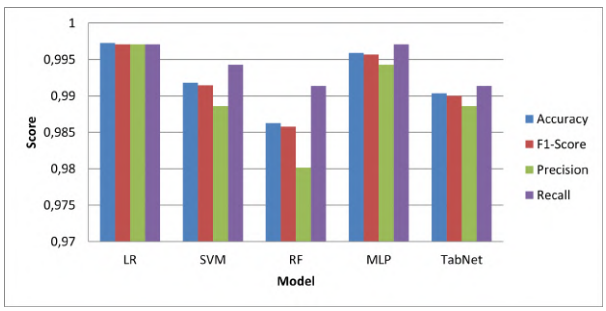


Fig. 3. Model Performance on MLRAN.

B. Model Performance on EldeRAN

Machine learning models were trained separately on the EldeRAN dataset to determine how they performed. Using EldeRAN, performance of the models was evaluated using different feature sizes of 483, 800 and 1,200 for each experiment to see how the number of features used effect model performance. Table III shows the results that were obtained by each traditional machine learning model using the EldeRAN dataset, while Table IV shows the results for the deep learning models for the different feature sizes.

Table III provides a summary of the traditional learning models across the three different feature sizes. RF achieved an accuracy of 0.9869, F1-score of 0.9829, precision of 0.9773 and a recall of 0.9885. These performance scores were maintained across the different feature sizes. This highlights how the model is highly sturdy to the addition of features beyond the initial selection of 483 as it identifies ransomware with very little to no variation. SVM consistently achieved the same performance scores as RF across all feature sizes. The lack of improvement shows that SVM is sturdy to the increase in the number of features.

Using 483 features, LR achieved an accuracy score of 0.9825. The F1-score, precision and recall are all at 0.9770. With additional features (800 and 1,200), the model had a slight increase in performance by 0,0044 in accuracy to 0.9869. F1 score saw a rise by 0.0057 to 0.9827. Precision went up by 0.0114 to 0.9884. Recall stayed constant at 0.9770 across the different feature sizes. The slight improvement in the accuracy, F1-score and precision at 800 and 1,200 features suggests that the model benefits from the additional feature information. Recall stayed consistent as the other metrics improved. This is due to additional features helping reduce false positives.. With additional information being provided, the precision of the model was higher than that of RF and SVM, with accuracy matching RF and SVM and F1-score slightly behind at 0.9827.

Recall is higher than precision for all models across the different feature sizes except for LR using 483 features. This implies that the models may prioritise identifying ransomware at the expense of increased false positives [27]. This shows that the models are calibrated to err on the side of caution, due to the consequence of a false negative being more catastrophic compared to flagging a benign file, which is a minor inconvenience through causing a false alarm.

TABLE III
TRADITIONAL MODELS ON ELDERAN

Model	Features	Accuracy	F1-Score	Precision	Recall
RF	483	**0.9869**	**0.9829**	**0.9773**	**0.9885**
	800	0.9869	0.9829	0.9773	0.9885
	1,200	0.9869	0.9829	0.9773	0.9885
LR	483	0.9825	0.9770	0.9770	0.9770
	800	0.9869	0.9827	0.9884	0.9770
	1,200	**0.9869**	**0.9827**	**0.9884**	**0.9770**
SVM	483	**0.9869**	**0.9829**	**0.9773**	**0.9885**
	800	0.9869	0.9829	0.9773	0.9885
	1,200	0.9869	0.9829	0.9773	0.9885

Fig 4 illustrates the effects of different feature sizes on the

performance of the traditional models based on the results of Table III. A key observation is the stability of traditional models. As illustrated in Fig 4, the precision for LR rises when the features increases from 483 to 800 features and remains constant as more features (1,200) are added. F1-score for LR also rises as additional information is provided from 483 to 800 features, but remains stagnant when more features are provided at 1,200. The accuracy for LR goes up when the number of features increases from 483 to 800, but stays constant when the features are 1,200.

Table IV provides a summary of the deep learning models across the three different feature sizes. Using 483 features, MLP had an accuracy score of 0.9738, and F1-score, precision and recall all balanced at 0.9655. With 800 features, the models show improvements in accuracy by 0.0087 to 0.9825, F1-score by 0.0112 to 0.9767, and precision by 0.0227 to 0.9882, while recall remains constant. This highlights that the initial feature set of 483 features was already effective in identifying the positive cases, with the additional features contributing to an increased precision to reduce false positives. Performance improvement continues with 1,200 features, with accuracy increasing by 0.0044 to 0.9825, F1 score rising by 0,006 to 0.9827, precision going up by 0.0002, and recall increasing by 0.0115 to 0.9770. The continuous improvement in performance shows that the model makes use of additional information to reduce the number of false positives while recall stays high.

TabNet results illustrate a decline in performance as the number of features goes up. With 483 features, the performance metrics are good with accuracy at 0.8690, F1-score at 0.8485, precision at 0.7568 and recall at 0.9655. All the performance drops as the number of features increases, with the lowest recorded using 1,200 features, at which accuracy is 0.5502, F1-score at 0.5863, precision at 0.4506 and recall at 0.8391. Recall remains relatively high, suggesting that the model may be overpredicting ransomware, leading to the very low precision score.

TABLE IV
DEEP MODELS ON ELDERAN

Model	Features	Accuracy	F1-Score	Precision	Recall
MLP	483	0.9738	0.9655	0.9655	0.9655
	800	0.9825	0.9767	0.9882	0.9655
	1,200	**0.9869**	**0.9827**	**0.9884**	**0.9770**
TabNet	483	**0.8690**	**0.8485**	**0.7568**	**0.9655**
	800	0.7817	0.7500	0.6637	0.8621
	1,200	0.5502	0.5863	0.4506	0.8391

Fig 5 visually represents the results of MLP shown in Table IV for the effect of feature selection. As illustrated, when feature size increases from 483 to 800, all the metrics rise, with only recall remaining constant at 0.9655. With a feature size of 1,200, all the performance scores continue to rise.

Complementing the TabNet results shown in Table IV, Fig 6 illustrates the effect of different features sizes. As illustrated, when feature size increases from 483 to 800 all the metrics have a drop which continues on as features go to 1,200.

Using Fig 5 and Fig 6, it can be seen that MLP and TabNet behave differently using different feature sizes. MLP's feed-forward structure allowed it to benefit from an increase in the number of features leading to better performance. Conversely, TabNet performances dropped as additional information was provided. This negative impact may have been caused by the presence of irrelevant features which affected the model's attention mechanism.

Comparisons using the EldeRAN highlight the strengths of the models used in this study. The original EldeRAN paper proposed a model using regularised logistic regression that scored an accuracy of 97.62% AND 96.34% recall. The study by [25] used the EldeRAN dataset to train a regularised logistic regression (with particle swarm optimisation) model to detect ransomware. The traditional models used for this study (RF, SVM and LR) outscored the accuracy obtained by [25] of 97.33%. The literature by [28] used an autoencoder which scored a high recall of 99% at the expense of accuracy at 93%.

C. Analysis of precision, recall and F1-score

Focusing on the recall, precision, and F1-score provides insights into how the models are behaving as compared to accuracy alone. RF and SVM have a high recall, meaning that almost all ransomware samples are identified. However, the recall is slightly higher than the precision, which shows a moderate number of false positives. Having a recall that is higher than precision means SVM and RF models are willing and more lenient to classify a sample as ransomware. When provided with additional information (as the number of features increases), MLP showcases an improvement in F1-score and precision. This highlights that MLP reduces the number of false alarms while demonstrating its ability to take advantage of non-linear feature interactions. TabNet showcased a decreasing score in precision, which shows the risk of false alarms when feature dimensions are not carefully managed.

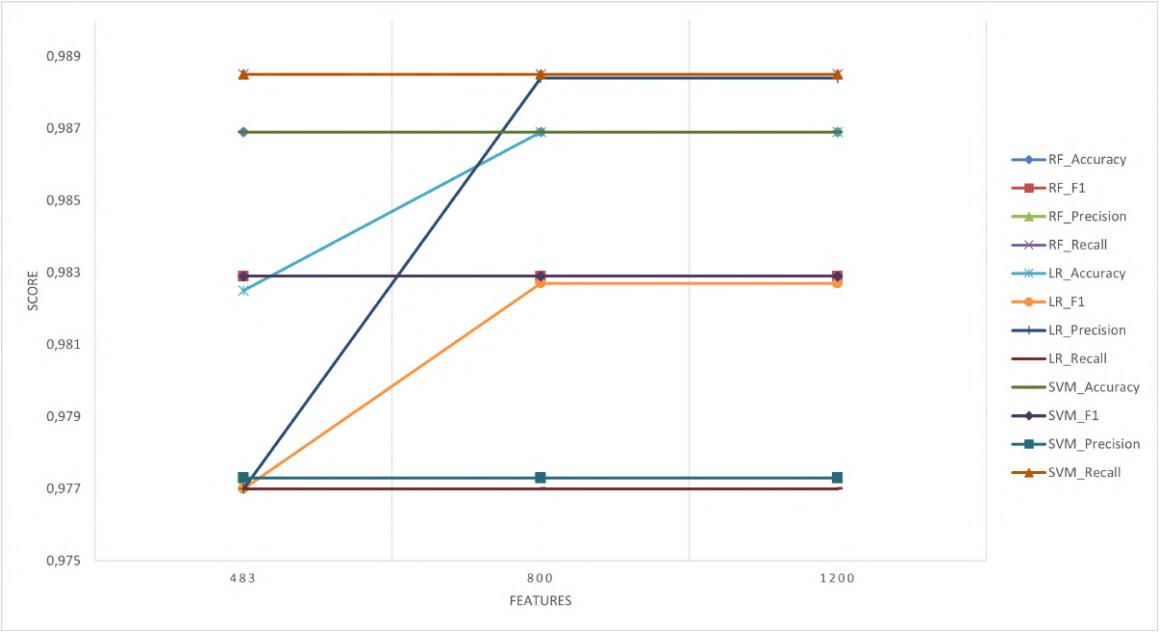


Fig. 4. Traditional Model Performance on EldeRAN using different feature sizes.

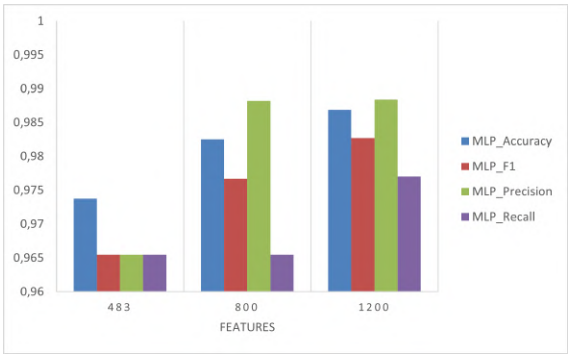


Fig. 5. MLP Performance on EldeRAN using different feature sizes.

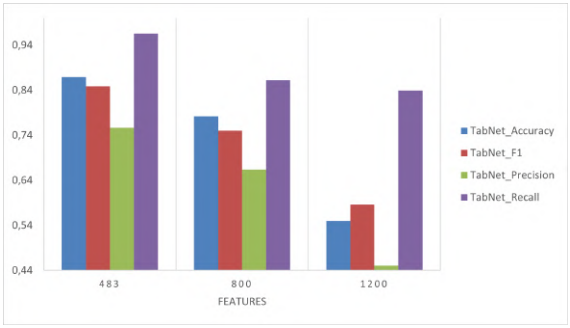


Fig. 6. TabNet Performance on EldeRAN using different feature sizes.

V. Discussion

This study aimed to compare the performance of deep learning and traditional machine learning models for ransomware detection and to determine the influence of feature selection techniques on the performance of these models. The use of the MLRan and EldeRAN dataset for evaluating the machine learning models used in this study provided several insights on the role played by feature dimensionality and model selection.

The results obtained in Section IV show that focusing on precision, recall, and F1-score gives better conclusions as compared to focusing on the accuracy of a model only. A notable observation from the results is that in almost all instances, the recall score outperformed the precision score for most of the models. This observation suggests that the models tend to give priority to identifying ransomware even at the expense of misclassifying some goodware as ransomware (generating some false positives). Failure to identify ransomware may cause severe damage to systems and infrastructure, as valuable data and information are rendered inaccessible. Therefore, while misclassifying legitimate software as ransomware is undesirable, it may only lead to additional efforts to verify the false alarm. Hence, RF and SVM, which achieved high recall and lower precision on EldeRAN and MLP and LR on MLRan, prioritise reducing the risks and costs associated with missing ransomware.

Precision must also not be overlooked. A detection tool that gives too many false alarms risks becoming unsuitable to deploy in real-world environments [29]. The balance between avoiding false alarms and identifying all ransomware is shown in the F1-score, which is the harmonic mean of the two metrics. Using the MLRan dataset, the best balance was achieved by LR with a score of 0.9971, highlighting that it had good recall and very few false positives. With the EldeRAN dataset, SVM and RF performed best with 0.9829 F1-score. MLP had the best balance at 0.9827 achieved with 1,200 features. The improved performance of MLP shows the potential of deep learning models to capitalise on additional information.

Another observation from the results is the stability shown by RF, LR and SVM. Besides strong performance metrics,

these models maintained the same scores at different feature sizes for EldeRAN. MLP is capable of matching or outperforming the traditional models in some instances, but requires careful hyperparameter tuning. The performance improvement as the number of features increases shows how the model benefits from additional information.

The evaluation results imply a trade-off between adaptability and simplicity. Traditional models gave a strong baseline performance showcasing minimal to no sensitivity in the number of features used. This makes them efficient and reliable for detection. On the other hand, deep models, particularly MLP, show promise when a large dataset with well-curated features is used, whereas TabNet's performance is reduced, showcasing the complexity that they bring.

VI. Conclusion and Future Work

This study successfully completed its objectives by demonstrating differences in model behaviour based on their architectures and on the number of features used. Five models, Logistic Regression, Random Forests, Support Vector Machine (linear kernel), Multi-layer Perceptron and TabNet were evaluated using accuracy, F1-score, precision and recall. Rather than focus on the accuracy of a model alone, the findings of the study suggest having more attention on precision, recall and F1-score, which give deeper insights into the trade-offs that arise when making use of these models. The findings of this study suggest that ransomware detection models must have a good balance between precision and recall, which was achieved by LR (accuracy - 0.9973, F1-score - 0.9971, precision - 0.9971, recall - 0.9971) and MLP (accuracy - 0.9959, F1-score - 0.9957, precision - 0.9943, recall - 0.9971). Overall, traditional learning models demonstrated sturdiness and minimal sensitivity to the number of features used as illustrated in Fig 4, which makes them reliable and consistent, making them attractive for ransomware detection.

While this study has given valuable insights into the performance of both traditional learning and deep learning models for ransomware detection using datasets with different feature dimensions, future work should focus on expanding this performance comparison to real-world unbalanced datasets where ransomware is less common and detection is more challenging. This will require exploration and use of imbalance handling methods such as SMOTE as a model's perfomance depends on how the data imbalance is handled during training. Adding on, different techniques for feature selection and reducing dimensionality can be explored to improve the performance of a model like TabNet. Lastly, future research may explore how boosting models perform to evaluate which models are best for real-world practical situations. These directions offer promising opportunities to improve the ransomware detection landscape.

Acknowledgements

The authors would like to thank [14], [16] for the publicly available datasets used in this study.

This research was undertaken in the Distributed Multimedia CoE at Rhodes University, with financial support from Telkom SA. The authors acknowledge that opinions, findings, conclusions, or recommendations expressed here are those of the author(s) and that none of the above-mentioned sponsors accept liability whatsoever in this regards.

The authors wish to acknowledge the use of AI writing tools, Writefull (integrated in Overleaf) and Grammarly. These tools assisted with punctuation, grammar, phrasing and academic writing style. The paper remains an accurate representation of the authors' underlying work and intellectual contributions.

References

[1] A. Arabo, R. Dijoux, T. Poulain, and G. Chevalier, "Detecting ransomware using process behavior analysis," *Procedia Computer Science*, vol. 168, pp. 289–296, 2020.

[2] H. Madani, N. Ouerdi, A. Boumesaoud, and A. Azizi, "Classification of ransomware using different types of neural networks," *Scientific Reports*, vol. 12, no. 1, p. 4770, 2022.

[3] J. Beaumarchais, P. Weber, N. Callaghan, and G. Bauer, "Integrated dynamic behavioral fingerprinting for real-time ransomware detection: A pattern-based classification," *OSF Preprints*, 2024.

[4] E. Anikolova, S. Martins, D. Rozental, J. Fontana, and P. Maier, "Ransomware detection through behavioral attack signatures evaluation: A novel machine learning framework for improved accuracy and robustness," *Authorea Preprints*, 2024.

[5] S. Kiyol, D. Franchini, E. Moreno, R. Kowalski, W. Fernandez, and A. Milosevic, "Ransomware detection using LSTM networks and file entropy analysis: A sequence-based approach," 2024, preprint.

[6] F. Mercaldo, "A framework for supporting ransomware detection and prevention based on hybrid analysis," *Journal of Computer Virology and Hacking Techniques*, vol. 17, no. 3, pp. 221–227, 2021.

[7] R. Anderson, C. Barton, R. Bölme, R. Clayton, C. Ganán, T. Grasso, M. Levi, T. Moore, and M. Vasek, "Measuring the changing cost of cybercrime," 2019.

[8] J. Hamill, A. Villareal, R. Costanzo, D. V. Dermeer, G. Ivanovich, and H. Macpherson, "Detecting ransomware via hybrid entropic behavior monitoring (hebm)," 2024, preprint.

[9] S. Kok, A. Abdullah, N. Jhanjhi, and M. Supramaniam, "Ransomware, threat and detection techniques: A review," *Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 2, p. 136, 2019.

[10] C. C. Moreira, C. S. de Sales Jr., and D. C. Moreira, "Understanding ransomware actions through behavioral feature analysis," *Journal of Communication and Information Systems*, vol. 37, no. 1, pp. 61–76, 2022.

[11] N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, "Cryptolock (and drop it): Stopping ransomware attacks on user data," in *Proc. IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2016, pp. 303–312.

[12] A. H. Celdrán, P. M. S. Sánchez, J. von der Assen, D. Shushack, Á. L. P. Gómez, G. Bovet, G. M. Pérez, and B. Stiller, "Behavioral fingerprinting to detect ransomware in resource-constrained devices," *Computers & Security*, vol. 135, p. 103510, 2023.

[13] E. Berrueta, D. Morato, E. Magaña, and M. Izal, "Crypto-ransomware detection using machine learning models in file-sharing network scenarios with encrypted traffic," *Expert Systems with Applications*, vol. 209, p. 118299, 2022.

[14] F. C. Onwuegbuche, A. Olaoluwa, A. D. Jurcut, and L. Pasquale, "Mlran: A behavioural dataset for ransomware analysis and detection," *arXiv preprint*, 2025, arXiv:2505.18613.

[15] Y. A. Ahmed, S. Huda, B. A. S. Al-Rimy, N. Alharbi, F. Saeed, F. A. Ghaleb, and I. M. Ali, "A weighted minimum redundancy maximum relevance technique for ransomware early detection in industrial iot," *Sustainability*, vol. 14, no. 3, p. 1231, 2022.

[16] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection," *arXiv preprint*, 2016, arXiv:1609.03020.

[17] A. Zheng and A. Casari, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly Media, Inc., 2018.

[18] M. Jemal, "Detection of crypto-ransomware attack using deep learning," 2023, unpublished.
[19] U. Urooj, B. A. S. Al-Rimy, A. Zainal, F. A. Ghaleb, and M. A. Rassam, "Ransomware detection using the dynamic analysis and machine learning: A survey and research directions," *Applied Sciences*, vol. 12, no. 1, p. 172, 2021.
[20] H. Monota and Y. Shigeta, "Optimizing alignment with progressively selective weight enhancement in large language models," 2024, unpublished.
[21] B. Xu and S. Wang, "Examining windows file system irp operations with machine learning for ransomware detection," 2024, preprint.
[22] N. Anderson, M. Li, and M. Evans, "Mapping the ransomware ecosystem: Tracing network traffic to command & control centers," 2024, preprint.
[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[24] W. Hardy, L. Chen, S. Hou, Y. Ye, and X. Li, "A deep learning framework for intelligent malware detection," in *Proc. Data Mining*, 2016, pp. 1–7.
[25] M. S. Abbasi, H. Al-Sahaf, M. Mansoori, and I. Welch, "Behavior-based ransomware classification: A particle swarm optimization wrapper-based approach for feature selection," *Applied Soft Computing*, vol. 121, p. 108744, 2022.
[26] S. Baramal and D. S. Chauhan, "A comparative study of tabnet and xgboost for tabular data classification," unpublished.
[27] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
[28] U. Zahoora, A. Khan, M. Rajarajan, S. H. Khan, M. Asam, and T. Jamal, "Ransomware detection using deep learning based unsupervised feature extraction and a cost sensitive pareto ensemble classifier," *Scientific Reports*, vol. 12, no. 1, p. 15647, 2022.
[29] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symposium on Security and Privacy*. IEEE, 2010, pp. 305–316.

Comparison of robustness of modulation classifiers against white-box adversarial attacks

Arnold van der Merwe
School of Electrical, Electronic and Computer Engineering
North-West University
Potchefstroom, South Africa
arnold1.vandermerwe@gmail.com

Albert Helberg
School of Electrical, Electronic and Computer Engineering
North-West University
Potchefstroom, South Africa
albert.helberg@nwu.ac.za

***Abstract*—Automatic modulation classification (AMC) is used in cognitive wireless radio systems to determine the modulation type of unknown signals without prior knowledge. AMC can be approached using likelihood-based methods, features-based methods, clustering, or more recently through deep neural networks (DNNs). The DNNs outperform the other approaches to AMC while being able to function in more complicated signal channels where the other AMC approaches cannot function. DNNs are, however, vulnerable to adversarial attacks that can greatly disrupt their classification performance. This paper compares the robustness of different classes of AMC techniques against white-box adversarial attacks in a noisy signal channel. The representative AMC techniques investigated are the parameter estimation and transformation-based CNN-GRU deep neural network (PET-CGDNN) for the deep learning (DL) based approach, the quasi-hybrid likelihood ratio test (QHLRT) for the likelihood-based approach, a k-nearest neighbour (KNN) using higher-order cumulants for the feature-based approach, and a clustering classifier for the clustering approach. The white-box adversarial attacks consist of the fast gradient method (FGM), projected gradient descent (PGD), and DeepFool, where the FGM and PGD attacks are fixed at a specific attack strength. The PET-CGDNN outperforms the other classifiers in the absence of attacks, but is the most affected classifier when attacks are added. Limited transferability of the white-box attacks generated on the PET-CGDNN to the other classifiers is shown.**

***Index Terms*—modulation classification, adversarial attacks, deep neural networks, k-nearest neighbours, likelihood ratio test**

I. Introduction

In wireless communications, deep neural networks (DNNs) are used for channel estimation, resource allocation, interference management, and spectrum sensing [1]–[3]. These DNNs typically outperform the more traditional approaches to these applications.

One of the fundamental problems in low-level wireless systems is automatic modulation classification (AMC), which aims to identify the modulation type of a received signal frame without prior knowledge. Signals are transmitted over wireless channels, which are subject to noise, fading, and interference, making robust AMC a challenging task and a focus of ongoing research [4]–[6].

Typically, AMC is approached using either likelihood-based techniques, features-based techniques, clustering techniques, or deep learning (DL)-based techniques. Recent research has focused on using DNNs as these DNNs are able to outperform the other approaches of AMC in complex signal channels.

The likelihood-based approach to AMC uses statistics to determine the modulation type of the received signal. In this approach, the classification task is framed as a hypothesis testing problem, where each modulation type corresponds to a hypothesis [7]. A likelihood function is used to determine the likelihood that the received signal is of a specific modulation type through the use of signal channel parameters and the received signal samples. The modulation type with the highest likelihood is identified as the one used through the application of the maximum likelihood criterion. The likelihood-based approach can be divided into three different categories depending on how any unknown signal channel parameters are handled, namely the average likelihood ratio test (ALRT), generalised likelihood ratio test (GLRT), and the hybrid likelihood ratio test (HLRT) [7], [8]. In the ALRT, unknown parameters are considered random variables and are averaged in the likelihood function, while the GLRT handles them as deterministic unknowns using maximum likelihood estimates of these parameters to compute the likelihood. Combining the ALRT and the GLRT results in the HLRT, which treats some parameters as random variables and others as deterministic unknowns.

The features-based approach to AMC consists of using the inherent characteristics of each modulation type for classification, typically through a supervised machine learning classifier such as k-nearest neighbour (KNN), decision tree, Naive Bayes, or support vector machine (SVM). These algorithms utilise features such as higher-order statistical moments and cumulants, wavelet features, cyclostationary analysis-based features, or signal spectral features in the form of amplitudes and frequencies [9]. Computing these features and using a machine learning algorithm has generally lower computational complexity than using one of the likelihood-based approaches, but provides a sub-optimal prediction if channel knowledge is available [9].

Clustering has also been used to predict the modulation type of received signals [10]. This is an unsupervised machine learning technique that groups data points in clusters that are close to each other using a specific distance metric such as the Euclidean distance, the Manhattan distance, or other distance

measures. Additionally, clustering can be either known-order where the number of clusters are already known or unknown-order where the algorithm must determine the appropriate number of clusters. In AMC, clustering is directly applied to the received signal frame where the number of groups corresponds to the modulation order. An additional decision-making process is required to determine which exact modulation type has been received after clustering has been performed on the signal frame. The clustering approach is used less than the other approaches, as it generally requires a higher signal-to-noise ratio (SNR) to be able to recognise the modulation type [10].

The DL-based approach to AMC consists of using a DNN that is able to automatically extract the relevant features to classify the modulation type by training on a dataset. The architectures that are often used for this task are convolutional neural networks (CNNs), recurrent neural networks (RNNs), hybrids of CNNs and RNNs, and transformers [11]–[13]. Since DNNs are trained on a representative dataset, they can typically recognise the modulation type despite the fact that the signal has been degraded due to fading, phase offset, pulse shaping, and other effects. Datasets that are commonly used are the RML2016.10a, RML2018.01a, RML22, and others, where each dataset contains signal frames of multiple modulation types at varying SNRs [6], [14], [15].

Despite the impressive results of DNNs in the domain of AMC, they are notably vulnerable to adversarial attacks that can severely decrease their performance [16]–[18]. Adversarial attacks are perturbations added to the input of the model, which are generally subtle compared to the original input, that causes the model to misclassify the input. These attacks come in the form of white-box, gray-box, or black-box attacks [19]. White-box attacks rely on full access to the model, including its parameters, training configuration, and inputs, unlike black-box attacks where this knowledge is unavailable. Gray-box attacks have partial knowledge of the model, its parameters, its inputs and training configuration.

In this paper, we do a comparative study of the robustness of different AMC classifiers against selected white-box adversarial attacks. Each classifier uses a different approach to AMC to determine to what extent the characteristics of the modulation types that they utilise in classification are affected. An additive white Gaussian noise (AWGN) channel is used to evaluate the classifiers. The following AMC classifiers were selected:

- Parameter estimation and transformation-based CNN-GRU deep neural network (PET-CGDNN) [20]
- Quasi-hybrid likelihood ratio test (QHLRT) [7], [8]
- KNN using higher-order cumulants [21]
- Clustering classifier using k-means [22]

These AMC classifiers will be tested against the following white-box attacks:

- Fast gradient method (FGM) [23]
- Projected gradient descent (PGD) [24]
- DeepFool [25]

The FGM and PGD attacks will be given a fixed attack size.

II. Background

A. System model

In the general AMC scenario depicted in Figure 1, a transmitter modulates an input signal with data before transmitting over a channel to the receiver. Various channel effects such as noise and fading are added to the transmitted signal. The receiver does not have prior knowledge of the modulation scheme used by the transmitter and uses the automatic modulation classifier to determine this. The demodulator uses the classified modulation scheme to demodulate the transmitted data.

In the above system, the received signal vector $\mathbf{r}$, where $\mathbf{r} = [r_1, r_2, \ldots, r_K]^T$ and K is the signal frame length, can be mathematically expressed as in (1) [8]. The transmitted signal vector $\mathbf{s}$, where $\mathbf{s} = [s_1, s_2, \ldots, s_K]^T$, is affected by the phase change ϕ and amplitude change α from the signal channel. Furthermore, the signal channel adds AWGN to the signal, where $\mathbf{n} = [n_1, n_2, \ldots, n_K]^T$. Noise is commonly described using SNR, which gives the ratio between the signal power and the noise power, and is measured using decibel (dB). Note that this model assumes that a single amplitude and phase change is applied to the frame of the received signal samples.

$$\mathbf{r} = \alpha e^{j\phi}\mathbf{s} + \mathbf{n} \tag{1}$$

Phase-shift keying (PSK) and quadrature amplitude modulation (QAM) are two digital signal modulation types that are commonly used to modulate a signal. PSK encodes information by altering the phase of a single carrier wave in discrete amounts. QAM differs by modifying amplitude and phase simultaneously using two carrier waves, one in-phase (I) wave and one quadrature (Q) wave that is 90° out of phase with the I component. Both modulation types use defined I/Q symbols to represent data, where the number of distinct symbols is the modulation order. A higher modulation order allows more data to be transmitted per symbol, which improves spectral efficiency but increases sensitivity to noise and other channel impairments.

Constellation diagrams can be used to visually represent a signal modulated using PSK or QAM, where the I component is the x-axis and the Q component is the y-axis. Each I/Q symbol is plotted on the diagram, forming groups or clusters, where the spread of the received symbols around a defined symbol is determined by the negative effects from the signal channel.

B. Related work

Recent research has focused on developing DL models for AMC that outperforms the other models already in the domain through either classification performance or computational efficiency. IC-AMCNet, a CNN-based architecture introduced by Hermawan *et al.*, aimed at achieving good classification performance and reduced latency for beyond 5G communications by incorporating dropout and Gaussian noise layers [26]. The PET-CGDNN is a hybrid DNN developed by Zhang *et al.*, which uses both CNN and RNN layers, for good

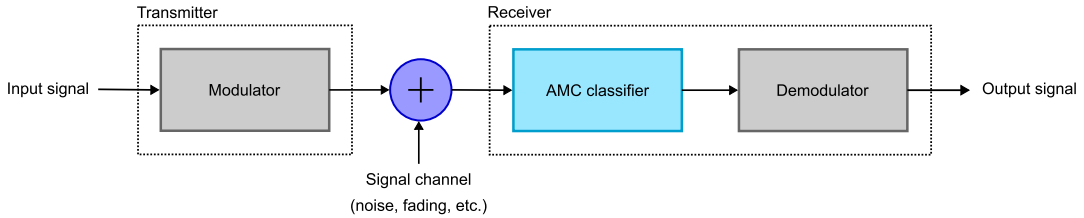


Fig. 1. Diagram showing simplified wireless communication system.

classification performance while being lightweight by having a low number of model parameters [20]. It also includes layers to estimate and correct the phase offset applied to a signal before attempting to classify the signal. Zheng *et al.* created the Ms-RaT model, which is a transformer-based model, that uses a dual-channel representation of a signal's amplitude and phase spectra and integrates a multi-scale pyramid module into its multilayer perceptron blocks [12]. It is able to outperform a CNN, a long short-term memory (LSTM), and a graph convolutional network (GCN).

Sadeghi and Larsson [17] tested white-box and black-box adversarial attacks on a DNN for AMC. Their research was some of the first research done on the effect adversarial attacks can have on DL-based methods for AMC. They used the RML2016.10a dataset and a CNN classifier (VT-CNN2). Their attack methods consisted of an altered FGM attack and the universal adversarial perturbation (UAP) attack. Their findings show that both white-box and black-box adversarial perturbations can significantly degrade classification accuracy with minimal added power. Adversarial attacks are significantly more effective than conventional jamming attacks, which highlights the vulnerability of DL-based methods for AMC.

Similar work was done by Lin *et al.* [27], where they investigated the vulnerability of the same model (VT-CNN2) and the same dataset (RML2016.10a) against white-box attacks. The fast gradient sign method (FGSM), PGD, basic iterative method (BIM), and momentum iterative method (MIM) attacks were selected to be used. They found that even imperceptible perturbations could reduce classification accuracy by 50% on average, with iterative attack methods being more effective than attack methods that took one step in generating the adversarial perturbation. Furthermore, they demonstrated that the waveform of the signal is visually unchanged after the perturbation has been added despite misclassification.

In [28], adversarial defense techniques were tested on a CNN for AMC. These defense techniques aim to make the DL-based classifier more robust against attacks altering the training or mechanism of the DNN. They evaluate the white-box FGSM, FGM, PGD, DeepFool, and Carlini & Wagner attacks, as well as the black-box UAP attack against a CNN classifier trained on the RML2016.10a dataset. The adversarial defense techniques of randomised smoothing, hybrid projected gradient descent adversarial training, and fast adversarial training were used. They found in their results that defense techniques can substantially improve robustness against attacks. The black-box attack of UAP is reduced to being only as harmful as a classical jamming attack, which requires a large amount of power. Despite the defense techniques, some vulnerability to white-box attacks still remains as the adversarial perturbation power approaches the noise power.

C. Classifier models

In this paper the PET-CGDNN, QHLRT, KNN with higher-order cumulants, and clustering classifiers were selected based on previous research in the AMC field. This section provides background information on how these classifiers function.

1) PET-CGDNN: For the DL approach to AMC, the lightweight PET-CGDNN model was selected to be used [20]. This is a hybrid DNN that uses CNN layers for spatial feature extraction and gated recurrent unit (GRU) layers for temporal feature extraction, using the advantages of both CNN and RNN architectures. Furthermore, it contains a module to estimate the phase offset that was applied to the received signal samples and a module to correct this phase offset through parametric inverse transformation. The PET-CGDNN model is able to achieve classification performance similar to other notable DNNs in AMC while having relatively few model parameters. Figure 2 provides the architecture of the PET-CGDNN model.

The model architecture is broken into three distinct parts, where each part has a different function [20].

Part 1 is used to estimate the phase offset applied to the received signal samples. The input to the model is a 2D vector of I and Q points, thus the first layer is used to flatten the input into a single dimension. A fully connected layer with a linear activation function estimates a phase offset parameter applied to each I/Q sample.

Part 2 applies the parametric inverse transformation through a custom layer described in (2). Each complex signal sample $r[n]$ is corrected by using an estimated phase offset $\hat{\varphi}$.

$$\hat{r}[n] = r[n]e^{-j\hat{\varphi}} = \begin{bmatrix} \Re\{r[n]\}\cos\hat{\varphi} + \Im\{r[n]\}\sin\hat{\varphi} \\ \Im\{r[n]\}\cos\hat{\varphi} - \Re\{r[n]\}\sin\hat{\varphi} \end{bmatrix} \quad (2)$$

The final part, Part 3, contains the layers used for classifying the modulation type of the received signal. First, the input is passed through a 2D convolutional layer with 75 filters and a kernel size of (2×8), followed by a rectified linear unit (ReLU) layer. Then another 2D convolutional layer is used with 25 filters and a kernel size of (1×5), followed by another ReLU layer. All spatial features were extracted with

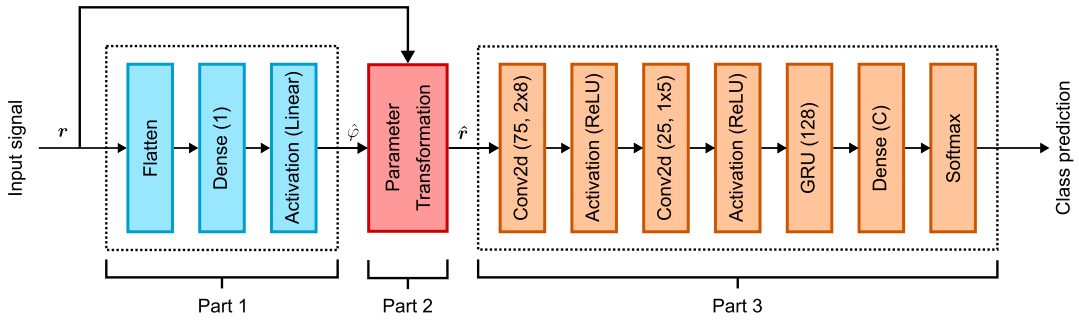


Fig. 2. PET-CGDNN architecture. Adapted from: [20].

these convolutional layers. Afterwards, a GRU layer with a size of 128 is used for temporal feature extraction, followed by a fully connected layer for classification. The size of the fully connected layer is determined by the number of modulation types in the dataset. The predicted modulation type is given by the final softmax layer.

2) QHLRT: The QHLRT variant of the HLRT was selected to be used for the likelihood-based approach to AMC [7], [8]. Like with the HLRT, the QHLRT uses a likelihood function to determine the most likely modulation type for the received signal samples with estimates of unknown channel parameters. Where the HLRT used maximum likelihood (ML) estimates of unknown signal channel parameters, the QHLRT uses method-of-moments (MoM) estimators which significantly reduces computational complexity. Specifically, the channel amplitude, phase, and noise is estimated using MoM estimators. The QHLRT is simple to implement, while having performance close to the HLRT [8].

The MoM estimators use moments for the estimation of the channel parameters. Moments can be calculated using (3), where M_{pq} denotes the (p, q)-th order moment. As defined previously in (1), $\boldsymbol{r}$ is the complex signal, where $\boldsymbol{r}^*$ is then the complex conjugate of $\boldsymbol{r}$ and $E[\cdot]$ is the expected or average statistical value.

$$M_{pq} = E[\boldsymbol{r}^{p-q}\boldsymbol{r}^{*q}] \tag{3}$$

The parameter $b^{(i)}$ is used in the various estimates of the channel parameters, where it differs for each distinct modulation type. It can be computed as follows, where $s_k^{(i)}$ are the noiseless signal constellation points of a distinct modulation type [8].

$$b^{(i)} = \frac{E[|s_k^{(i)}|^4]}{\left(E[|s_k^{(i)}|^2]\right)^2} \tag{4}$$

The parameter $b^{(i)}$ was determined for the PSK and QAM modulation types, with unit average power being used for rectangular QAM. The values for the different modulation types are provided in Table I.

If the $b^{(i)}$ parameter has been obtained, then it can be used to estimate the channel amplitude α, phase φ and noise N

TABLE I
PARAMETER $b^{(i)}$ FOR DIFFERENT MODULATION ORDERS OF PSK AND QAM.

Modulation type	**Value for $b(i)$**
M-PSK	1
8-QAM	1.444
16-QAM	1.32
32-QAM	1.31
64-QAM	1.381
128-QAM	1.342

using MoM estimators as given in the following equations [8]. The estimated moments $\hat{M}_{pq}$ are used with the modulation order M_i. All of these channel parameters are estimated for each expected modulation type. Two different equations are used when estimating the channel phase for a PSK signal or a QAM signal.

$$\hat{\alpha}^{(i)} = \left(\frac{\hat{M}_{42} - 2\hat{M}_{21}^2}{b^{(i)} - 2}\right)^{1/4} \left(E[|s_k^{(i)}|^2]\right)^{-1/2} \tag{5}$$

$$\hat{N}^{(i)} = \hat{M}_{21} - \left(\frac{\hat{M}_{42} - 2\hat{M}_{21}^2}{b^{(i)} - 2}\right)^{1/2} \tag{6}$$

$$\hat{\phi}^{(i)}_{\text{M-PSK}} = M_i^{-1}\arg\left(\sum_{k=1}^{K} r_k^{M_i}\right) \tag{7}$$

$$\hat{\phi}^{(i)}_{\text{M-QAM}} = 4^{-1}\arg\left(-\sum_{k=1}^{K} r_k^4\right) \tag{8}$$

When the channel parameters have been estimated, then the likelihood $\text{LF}^{(i)}$ of the received signal samples belonging to a modulation type can be obtained as in (10) [8]. The likelihood is determined through averaging the difference between the received signal symbols r_k and a modulation type's signal constellation points s_m. Notice that as in (1), the channel parameters are assumed to be constant over the interval that the signal symbols are received. The received signal is identified as the modulation type that has the highest likelihood, as calculated in (10).

$$\log\left(\mathcal{L}^{(i)}(\mathbf{r})\right) = \sum_{k=1}^{K} \log\left(\sum_{m=1}^{M_i} \exp\left(-\frac{1}{\hat{N}^{(i)}}\left|r_k - \hat{\alpha}e^{j\hat{\phi}}s_m\right|^2\right)\right) - \log\left(M_i \pi \hat{N}^{(i)}\right) \tag{9}$$

$$\mathcal{L}^{(i)}(\mathbf{r}) = \prod_{k=1}^{K} \frac{1}{M_i} \sum_{m=1}^{M_i} \frac{1}{\pi \hat{N}^{(i)}} \exp\left(-\frac{1}{\hat{N}^{(i)}}\left|r_k - \hat{\alpha}e^{j\hat{\phi}}s_m\right|^2\right) \tag{10}$$

Equation (10) is not practical to implement due to its computational complexity, thus the log-likelihood format of the equation is used as given in (9). The log-likelihood format of the equation also returns the highest likelihood for the correct modulation type.

Note that in this paper, only the noise parameter is estimated and that the channel amplitude is fixed at 1 and the channel phase offset is fixed at 0°. This was done to make a fair comparison with other AMC classifiers.

3) KNN with higher-order cumulants: Higher-order cumulants are one of the features frequently used for AMC [9], [21]. These cumulants are statistical features that describe the structure of a signal. Cumulant values vary with different modulation types which enables separation between modulation types as, for example, 8-PSK will produce different cumulant values compared to 16-QAM. Higher-order cumulants, fourth-order and above, are often used as they are relatively robust against AWGN.

Moments, calculated using (3), can be used to derive the different cumulants. For an example, the equations to compute the fourth order cumulants C_{40}, C_{41}, and C_{42} are provided below [21]. In the interest of brevity, the equations for the other order cumulants can be found in [21].

$$C_{40} = M_{40} - 3M_{20}^2 \tag{11}$$

$$C_{41} = M_{40} - 3M_{20}M_{21} \tag{12}$$

$$C_{42} = M_{42} - 2M_{21} - |M_{21}|^2 \tag{13}$$

A KNN can be paired with these higher-order cumulants to act as a classifier. A KNN is a supervised machine learning algorithm that assigns a class label to an unknown signal by comparing its feature vector to those of labelled training signals, where in this instance the feature vector consists of cumulants. The classification is performed through a majority vote from the k closest neighbours, using a specific distance metric such as Euclidean distance. The performance of the KNN is dependent on how well the used features are able to seperate the different modulation types.

4) Clustering: A clustering classifier from [22] was selected to be used as another approach to AMC. The general process of the clustering classifier starts with using a clustering algorithm for each expected modulation order in the dataset, where the modulation order corresponds to the number of expected clusters. The fit of the clusters to the received I/Q samples is evaluated using a scoring metric. The best scoring number of clusters is chosen as the identified modulation order. The clustering classifier then uses knowledge of the modulation types in the dataset to derive the modulation type. Generally, the modulation order between the PSK and QAM modulation types differs, where higher modulation orders are used by QAM, but there is some overlap especially with the modulation order of 8. In this case, the distances between the cluster centroids are evaluated to determine if the signal is PSK or QAM.

In [22], the well-known k-means clustering algorithm [29] was selected to be used in the clustering classifier due to its simplicity. The k-means algorithm uses a predefined number of clusters k, with the idea of grouping data points together such that data points within the same cluster are closer to each other than those in other clusters.

To determine the quality of the fit of the clusters, the Davies-Bouldin index (DBI) [30] is used as a scoring metric. The DBI is based on two quantities, the intra-cluster distance within each cluster and the inter-cluster separation. It uses these two quantities to calculate a score for the fit of the clusters, where a lower score means that the number of clusters is better suited for the received I/Q samples than other numbers of clusters.

The clustering classifier had to be able to distinguish between PSK and QAM when their modulation orders overlap in the dataset [22]. In the dataset used by [22], both 8-PSK and 8-QAM were present. They used the constellation shapes to differentiate between the two modulation schemes. When examining the constellation diagram of a PSK signal versus a QAM signal, it is evident that a PSK signal forms a circle around the origin while QAM signals form a grid. Refer to Figures 8 and 9 in Section IV for an example of the constellation diagram of 8-PSK and 16-QAM. For a sufficiently high SNR the cluster centroids will form a circle around the origin of the constellation diagram in the case of a received PSK modulated signal. The maximum distance and the minimum Euclidean distance from the origin to the centroids can be obtained and the difference between these two distances used as a deciding factor. If the difference is below a predefined threshold, the received signal is assumed to be a PSK modulated signal, while if it is above the threshold it assumed to be a QAM signal.

The required differentiating threshold was calculated to be 0.3568 [22]. This was obtained by examining an 8-QAM constellation diagram with unit average power. The Euclidean distances of the nearest and farthest symbols from the origin were measured, and their difference was computed to estimate the midpoint distance. This difference was halved to identify if it corresponds to PSK or QAM. To avoid misclassification,

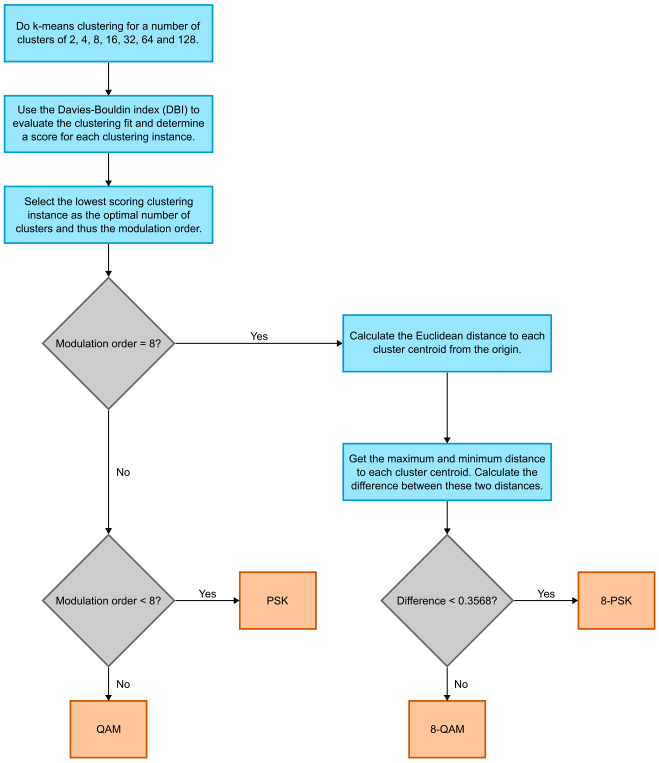


Fig. 3. Decision-making process for clustering classifier. Adapted from: [22]

the threshold was set at 97.27% of this halved value [22].

Figure 3 illustrates the clustering classifier process in a flowchart.

D. Adversarial attacks

1) Fast gradient method: The FGM is a white-box adversarial attack technique that perturbs inputs by moving them in the direction of the gradient of the DL model's loss function, with respect to the input [23]. Specifically, a fixed step size is taken against the gradient of the model's loss function, making the classifier more uncertain about the modulation type, potentially causing it to misclassify the input. It is a generalisation of the commonly known FGSM by rather being based on the L_2 norm, instead of the L_∞ norm [31].

The FGM is mathematically represented by (14), where $\boldsymbol{x}$ denotes the input, ϵ is a scaling factor, L represents the model's loss function, $\boldsymbol{\theta}$ are the model parameters, and y is the true class label [23]. As ϵ increases, the attack becomes more significant, increasing the likelihood of input misclassification.

$$\boldsymbol{x}_{adversarial} = \boldsymbol{x} + \epsilon \cdot \frac{\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \boldsymbol{x}, y)}{\|\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \boldsymbol{x}, y)\|_2} \tag{14}$$

2) Projected gradient descent: The PGD is another white-box adversarial attack that extends the approach used by the FGM, by taking multiple steps against the gradient instead of a single step [24]. Unlike the FGM, the PGD can refine the added perturbation over multiple iterations, making it more effective in finding attacks that degrade classifier performance while staying imperceptible in terms of signal quality. The attack size is still constrained by a scaling factor, ϵ, which can be visualised as a ball around the input with the radius ϵ in the case of the L_2 norm.

The equation for PGD attack when using the L_2 norm is provided as follows. The variable $\boldsymbol{x}^{t+1}$ represents the updated input with added perturbation, $\boldsymbol{x}^t$ denotes the current input, potentially already perturbed, and $\prod_{\boldsymbol{x}+S_\epsilon}$ acts as the projection operator limiting attack strength. Here, α denotes the attack step size, while other variables are the same as in (14) for the FGM attack [24]. The number of attack steps is usually predefined.

$$\boldsymbol{x}^{t+1} = \prod_{\boldsymbol{x}+S_\epsilon} \left(\boldsymbol{x}^t + \alpha \cdot \frac{\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \boldsymbol{x}, y)}{\|\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \boldsymbol{x}, y)\|_2} \right) \tag{15}$$

3) DeepFool: Where the other white-box adversarial attacks operated directly on the gradient of the model's loss function, the DeepFool attack operates on the model's decision boundaries [25]. DeepFool finds the minimal perturbation required to change the predicted modulation type of a signal by locally linearising the DNN's decision boundaries and projecting the input across the closest boundary in successive iterations, adding the perturbations as it goes along. This process continues until the prediction of the DNN changes.

In the case of multi-class classification, the closest hyperplane, which is the linearised decision boundary, can be found by the following equation [25]. Here, $\mathbf{x}_0$ is the original input, $\mathbf{w}$ is the computed gradient of the loss function and f is the class label of an input. The subscript $\hat{k}(\mathbf{x}_0)$ indicates the estimated label of the original input, whereas the subscript k is the true label.

$$\hat{l}(\mathbf{x}_0) = \arg \min_{k \neq \hat{k}(\mathbf{x}_0)} \frac{|f_k(\mathbf{x}_0) - f_{\hat{k}(\mathbf{x}_0)}(\mathbf{x}_0)|}{\|\mathbf{w}_k - \mathbf{w}_{\hat{k}(\mathbf{x}_0)}\|_2} \tag{16}$$

When the closest hyperplane has been found, the minimal perturbation $\mathbf{r}_*(\mathbf{x}_0)$, which causes the original input to be projected on the hyperplane indexed by $\hat{l}(\mathbf{x}_0)$, can be determined as follows [25]. This perturbation is added to the original input to create an attacked signal.

$$\mathbf{r}_*(\mathbf{x}_0) = \frac{|f_{\hat{l}(\mathbf{x}_0)}(\mathbf{x}_0) - f_{\hat{k}(\mathbf{x}_0)}(\mathbf{x}_0)|}{\|\mathbf{w}_{\hat{l}(\mathbf{x}_0)} - \mathbf{w}_{\hat{k}(\mathbf{x}_0)}\|_2^2} \left(\mathbf{w}_{\hat{l}(\mathbf{x}_0)} - \mathbf{w}_{\hat{k}(\mathbf{x}_0)} \right) \tag{17}$$

III. Methodology

In this section, the synthetic dataset used by the different modulation classifiers for evaluation is discussed, the training setup of the PET-CGDNN and the KNN is provided. Furthermore, the generation of adversarial attacks on the dataset is discussed in addition to their configuration.

TABLE II
HYPERPARAMETER TUNING RESULTS FOR PET-CGDNN.

Hyperparameter	Search space boundaries	Optimised value
Initial LR	$[1 \times 10^{-4}, 1 \times 10^{-1}]$	3.04×10^{-4}
Batch size	$[32, 512]$	438
Reduce LR patience	$[3, 20]$	8
Reduce LR factor	$[0.01, 0.9]$	0.8074

A. Dataset

A dataset was generated using MATLAB R2023b to compare the different classifiers for AMC in a signal channel affected by AWGN. The dataset contained the modulation types of BPSK, QPSK, 8-PSK, 8-QAM, 16-QAM, 32-QAM, 64-QAM, and 128-QAM, where unit average power was used for the QAM modulated signals. The SNR varied from -30 to 30 dB in intervals of 2 dB. At each SNR, random data was used to generate 2000 signal frames for each modulation type, where each frame had a length of 2048 I/Q symbols. The clustering classifier's performance is strongly dependent on the frame length, thus a sufficient frame length was selected for a fair comparison between the classifiers.

From the generated signal frames, a training, validation, and test set was randomly selected while maintaining class balance, so that there was an equal representation of each modulation type in each set. The ratio of 6:2:2 was used when splitting the frames for the training, validation, and test set. A third of the test set was selected, creating a reduced test set for the clustering classifier as the algorithm is computationally expensive, to speed up repeated experiments. This reduced dataset had no detrimental effect on the classification performance of the clustering classifier.

B. Training configuration

1) PET-CGDNN: The Keras framework alongside TensorFlow was used to implement, train, and evaluate the PET-CGDNN model. The model was trained using the generated training set and hyperparameter tuned using the validation set. An Adam optimiser paired with a learning rate (LR) scheduler decreased the LR when the validation loss plateaued. The epochs were restricted to 500, using early stopping if no improvement in validation loss occurred after 30 epochs. The model with the lowest validation loss was retained as the final version.

Optuna, a framework for hyperparameter optimisation using the tree-structured Parzen estimator (TPE) Bayesian optimisation algorithm, was utilised for hyperparameter tuning [32]. The initial LR, batch size, LR scheduler's patience and the LR scheduler's factor were optimised using Optuna. The trial with the minimum validation set loss was chosen and its parameters used. Table II provides the results of the best trial.

2) KNN with cumulants: The KNN used 5 neighbours ($K = 5$) for classification. The feature space was built from the complete training set, ignoring SNR, so that the KNN has complete representation of the dataset regardless of the SNR. Higher-order cumulants C_{40}, C_{41}, C_{42}, C_{60}, C_{61}, C_{62}, and C_{63} were used as features.

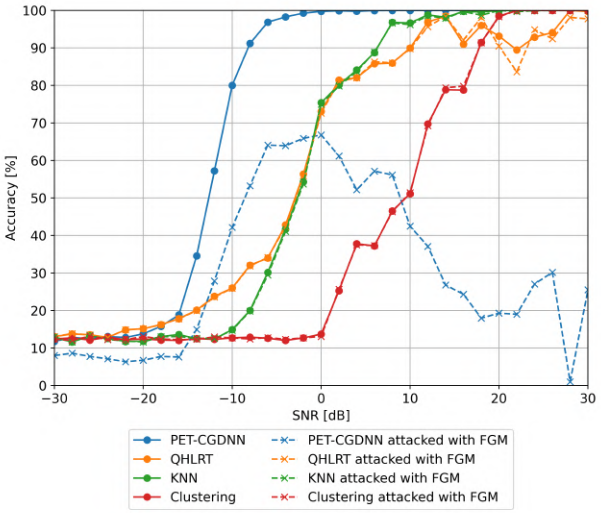


Fig. 4. Performance of classifiers on unperturbed and FGM attacked signals.

C. Generation of adversarial attacks

The white-box adversarial attacks of FGM, PGD and DeepFool were used to generate attacks on the test set using the trained PET-CGDNN model. The FGM and PGD attacks were configured to specifically use the L_2 norm and use an ϵ of 1.5. This gives these attacks a fixed size irrespective of the modulation type or SNR. Attack step size was configured to be 0.1 for the PGD attack method.

IV. RESULTS & DISCUSSION

The selected AMC classifiers were evaluated on the test sets before any attacks and after attacks had been added to determine their relative performance and robustness. Their classification accuracy at each SNR was captured.

Figure 4 shows the performance of the classifiers on the unperturbed signals and the signals attacked with the FGM attack. Examining the unperturbed performance, it is evident that the PET-CGDNN outperforms the other approaches to AMC and that the clustering classifier is the worst performing approach. The KNN classifier outperforms the QHLRT classifier as the QHLRT struggles at high SNRs with classifying nested QAM constellations, for example between 64-QAM and 16-QAM.

When the FGM attack is added, the performance of the QHLRT classifier, KNN classifier and the clustering classifier is mostly unaffected, whereas the classification performance of the PET-CGDNN is deteriorates significantly. Despite having a fixed attack size ($\epsilon = 1.5$), it is noticeable that the attack becomes more potent as the SNR increases. At lower SNRs the attack is less effective, potentially because the noise obfuscates the FGM attack.

The performance against the PGD attack is shown in Figure 5. The classification performance of the PET-CGDNN is more affected by the PGD attack than the FGM attack, even with

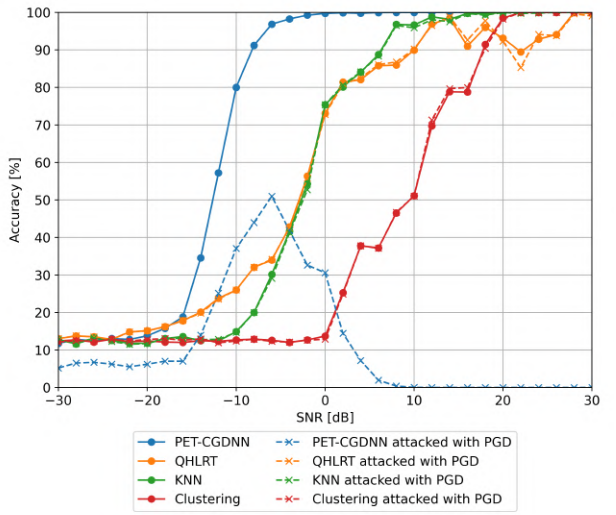


Fig. 5. Performance of classifiers on unperturbed and PGD attacked signals.

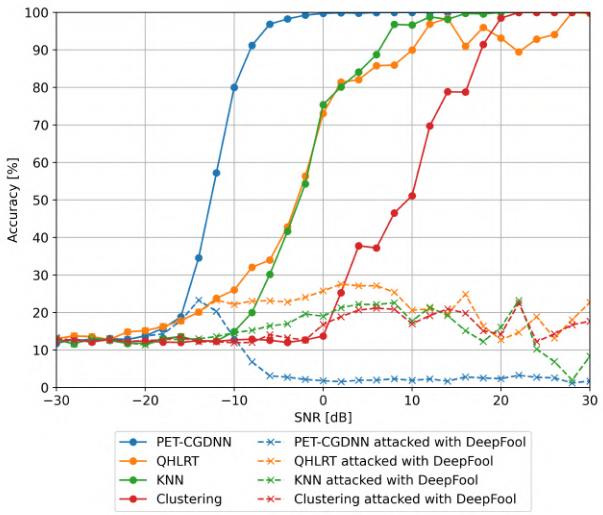


Fig. 6. Performance of classifiers on unperturbed and DeepFool attacked signals.

an attack of the same size $\epsilon = 1.5$. This is due to the multi-step nature of the PGD attack, which is able to generate a more fine-tuned attack against the gradient of the model's loss function. In Section II-B it was discussed that iterative attacks outperformed single-step attacks [27] and the same supporting result has been obtained in this paper. Similarly to the above FGM attack, the other classifiers are largely unaffected by the PGD attack. Both the FGM and PGD attacks show limited transferability from the PET-CGDNN to the other classifiers.

Figure 6 shows the performance against the DeepFool adversarial attack. With this attack, the classification performance of all of the classifiers are significantly affected, but the QHLRT, KNN and clustering classifier show more robustness against the attack than the PET-CGDNN. The PET-CGDNN's performance is reduced to be worse than random guessing as with the generated dataset random guessing would result in a classification accuracy of 12.5%.

A confusion matrix was generated to examine the performance of the PET-CGDNN against the FGM attacks at a SNR of 20 dB, which is provided in Figure 7. This diagram gives the classification accuracy for each modulation type and shows what each modulation type was misclassified as. For a classification accuracy of 100%, a diagonal line would be shown where each block has a value of 1.00. The confusion matrix in Figure 7 shows that most modulation types are significantly misclassified. The classification of the BPSK modulation type has the most robustness against the FGM attack, which is potentially due to its simplicity compared to the other modulation types.

The unperturbed and attacked constellation diagrams for 8-PSK and 16-QAM at a SNR of 20 dB were plotted to examine how obvious the attack is, examining it through the constellation diagram. The constellation diagrams are provided in Figures 8 and 9. Both the FGM and PGD attacked signal frames are mostly indistinguishable from the unperturbed

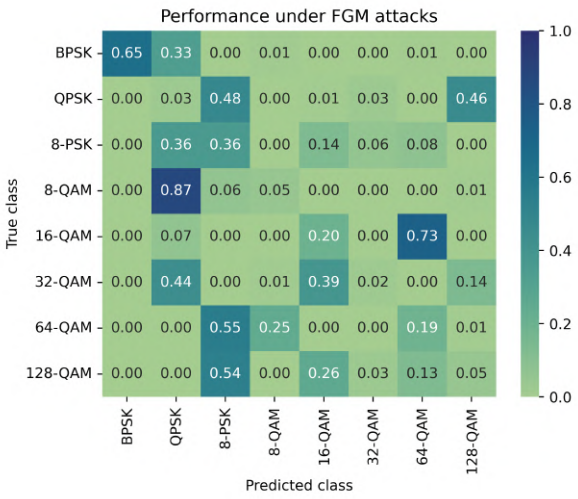


Fig. 7. Confusion matrix of PET-CGDNN against FGM attacks at SNR of 20 dB.

signal frame. Surprisingly, the DeepFool attack greatly disrupts the signal frame, significantly more than expected. The DeepFool attack determines the minimal perturbation required to go over the nearest decision boundary, but in this scenario some factor is causing it to have to significantly change the signal to cross the decision boundary. This can potentially be due to the DeepFool linearising the decision boundaries or the decision boundaries not being as expected.

V. Conclusion

The AMC classifiers of PET-CGDNN, QHLRT, KNN and clustering were evaluated against the white-box adversarial attacks of FGM, PGD and DeepFool. The FGM and PGD attacks were fixed at a specific attack strength. When using

FGM and PGD attacks, the classification performance of the PET-CGDNN was greatly reduced, while the QHLRT, KNN, and clustering classifiers showed limited change in their classification performance. Attacks using DeepFool significantly affected all of the classifiers, but the PET-CGDNN was the most affected classifier as its performance was reduced to worse than random guessing. These findings indicate that selected white-box attacks have limited transferability to the features used by classifiers of QHLRT, KNN using higher-order cumulants, and clustering, if the attack size is restricted.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support of this study by the Telkom CoE at the NWU.

REFERENCES

[1] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep Learning-Based Channel Estimation," *IEEE Communications Letters*, vol. 23, no. 4, pp. 652–655, Apr. 2019.

[2] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[3] Y. Zhang and Z. Luo, "A Review of Research on Spectrum Sensing Based on Deep Learning," *Electronics*, vol. 12, no. 21, p. 4514, Jan. 2023.

[4] Y. Wang, G. Gui, T. Ohtsuki, and F. Adachi, "Multi-Task Learning for Generalized Automatic Modulation Classification Under Non-Gaussian Noise With Varying SNR Conditions," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3587–3596, Jun. 2021.

[5] Q. Zheng, X. Tian, Z. Yu, M. Yang, A. Elhanashi, and S. Saponara, "Robust automatic modulation classification using asymmetric trilinear attention net with noisy activation function," *Engineering Applications of Artificial Intelligence*, vol. 141, p. 109861, Feb. 2025.

[6] V. Sathyanarayanan, P. Gerstoft, and A. E. Gamal, "RML22: Realistic Dataset Generation for Wireless Modulation Classification," *IEEE Transactions on Wireless Communications*, vol. 22, no. 11, pp. 7663–7675, Nov. 2023.

[7] F. Hameed, O. A. Dobre, and D. C. Popescu, "On the likelihood-based approach to modulation classification," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 5884–5892, Dec. 2009.

[8] O. A. Dobre and F. Hameed, "Likelihood-Based Algorithms for Linear Digital Modulation Classification in Fading Channels," in *2006 Canadian Conference on Electrical and Computer Engineering*, May 2006, pp. 1347–1350.

[9] Z. Zhu and A. K. Nandi, *Automatic Modulation Classification: Principles, Algorithms and Applications*. Wiley, 2015.

[10] J. P. Mouton, M. Ferreira, and A. S. J. Helberg, "A comparison of clustering algorithms for automatic modulation classification," *Expert Systems with Applications*, vol. 151, p. 113317, Aug. 2020.

[11] F. Zhang, C. Luo, J. Xu, Y. Luo, and F.-C. Zheng, "Deep learning based automatic modulation recognition: Models, datasets, and challenges," *Digital Signal Processing*, vol. 129, p. 103650, Sep. 2022.

[12] Q. Zheng, P. Zhao, H. Wang, A. Elhanashi, and S. Saponara, "Fine-Grained Modulation Classification Using Multi-Scale Radio Transformer With Dual-Channel Representation," *IEEE Communications Letters*, vol. 26, no. 6, pp. 1298–1302, Jun. 2022.

[13] J. Cai, F. Gan, X. Cao, and W. Liu, "Signal Modulation Classification Based on the Transformer Network," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 3, pp. 1348–1357, Sep. 2022.

[14] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional Radio Modulation Recognition Networks," in *Engineering Applications of Neural Networks*. Springer International Publishing, 2016, pp. 213–226.

[15] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-Air Deep Learning Based Radio Signal Classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, Feb. 2018.

[16] M. Usama, M. Asim, J. Qadir, A. Al-Fuqaha, and M. A. Imran, "Adversarial Machine Learning Attack on Modulation Classification," in *2019 UK/ China Emerging Technologies (UCET)*, Aug. 2019, pp. 1–4.

[17] M. Sadeghi and E. G. Larsson, "Adversarial Attacks on Deep-Learning Based Radio Signal Classification," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 213–216, Feb. 2019.

[18] W. Zhang, M. Krunz, and G. Ditzler, "Stealthy Adversarial Attacks on Machine Learning-Based Classifiers of Wireless Signals," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 261–279, 2024.

[19] Y. Wang, T. Sun, S. Li, X. Yuan, W. Ni, E. Hossain, and H. Vincent Poor, "Adversarial Attacks and Defenses in Machine Learning-Empowered Communication Systems and Networks: A Contemporary Survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2245–2298, 2023.

[20] F. Zhang, C. Luo, J. Xu, and Y. Luo, "An Efficient Deep Learning Model for Automatic Modulation Recognition Based on Parameter Estimation and Transformation," *IEEE Communications Letters*, vol. 25, no. 10, pp. 3287–3290, Oct. 2021.

[21] M. Venkata Subbarao and P. Samundiswary, "Performance Analysis of Modulation Recognition in Multipath Fading Channels using Pattern Recognition Classifiers," *Wireless Pers Commun*, vol. 115, no. 1, pp. 129–151, Nov. 2020.

[22] A. van der Merwe and A. Helberg, "Modulation classification accuracy of a clustering and a CNN technique under FGSM adversarial attacks," in *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*, 2024, pp. 197–203.

[23] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting Adversarial Attacks with Momentum," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 9185–9193.

[24] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," in *2018 International Conference on Learning Representations*, Feb. 2018.

[25] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582, Jun. 2016.

[26] A. P. Hermawan, R. R. Ginanjar, D.-S. Kim, and J.-M. Lee, "CNN-Based Automatic Modulation Classification for Beyond 5G Communications," *IEEE Communications Letters*, vol. 24, no. 5, pp. 1038–1041, May 2020.

[27] Y. Lin, H. Zhao, X. Ma, Y. Tu, and M. Wang, "Adversarial Attacks in Modulation Recognition With Convolutional Neural Networks," *IEEE Trans. Rel.*, vol. 70, no. 1, pp. 389–401, Mar. 2021.

[28] B. R. Manoj, P. M. Santos, M. Sadeghi, and E. G. Larsson, "Toward Robust Networks against Adversarial Attacks for Radio Signal Modulation Classification," in *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, Jul. 2022, pp. 1–5.

[29] D. Xu and Y. Tian, "A Comprehensive Survey of Clustering Algorithms," *Ann. Data. Sci.*, vol. 2, no. 2, pp. 165–193, Jun. 2015.

[30] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[31] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," Mar. 2015.

[32] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-Generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

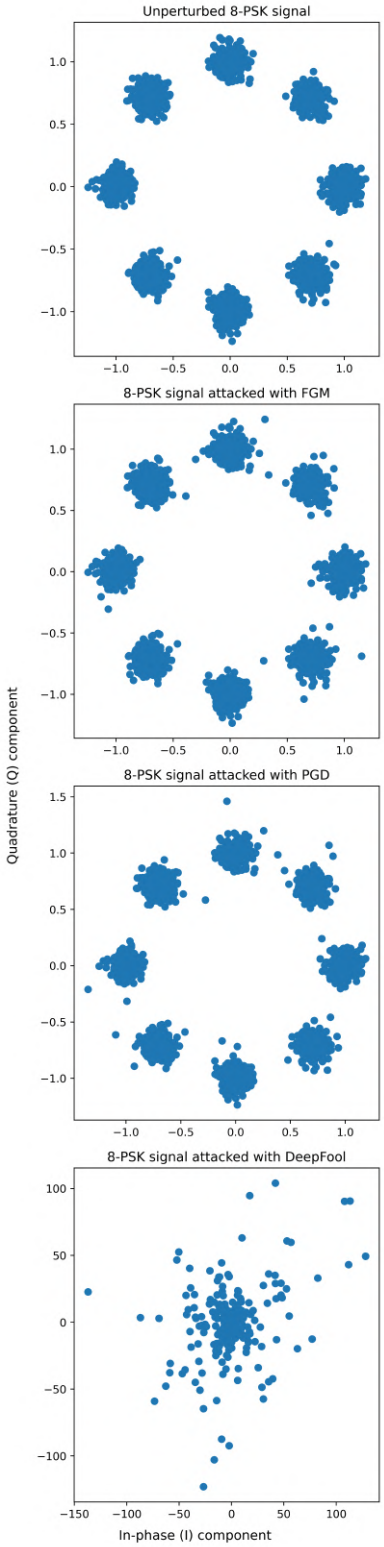


Fig. 8. Constellation diagrams of a 8-PSK signal frame unperturbed and attacked with different methods at 20 dB SNR.

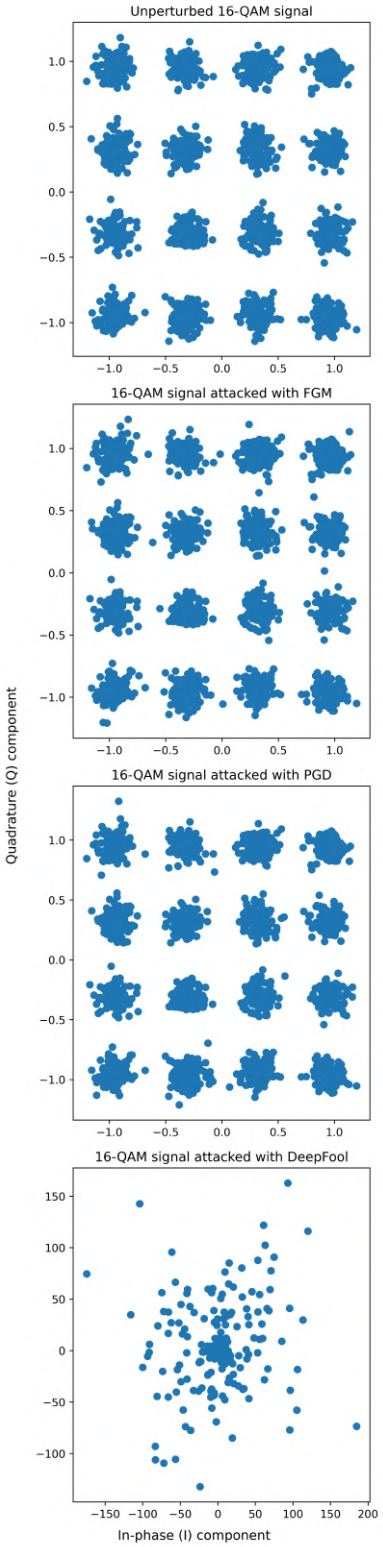


Fig. 9. Constellation diagrams of a 16-QAM signal frame unperturbed and attacked with different methods at 20 dB SNR.

Experimental Characterisation of an Erbium-Doped Fibre Amplified Spontaneous Emission Source for Sensing Systems

1st K. Josue KALOLO
Electrical and Electronic Engineering science
University of Johannesburg
Johannesburg, South Africa
irkalolojosh@gmail.com

2nd MF Grobler
Electrical and electronic engineering science
University of Johannesburg
Johannesburg, South Africa
michaelg@uj.ac.za

3rd JJM KABOKO
Electrical and electronic engineering science
University of Johannesburg
Johannesburg, South Africa
mongatvch@gmail.com

***Abstract*— This study presents a comprehensive experimental characterisation of an erbium-doped fibre amplified spontaneous emission (EDF-ASE) source, optimised for next-generation sensing applications. We systematically investigated the spectral dynamics, temporal stability, noise characteristics, and environmental sensitivity using both forward-pumped and bidirectional pumping architectures. Performance metrics suitable for precision optical sensing systems were achieved, including a spectral flatness of ±1.2 dB over a 35 nm bandwidth, an output power of up to -2.84 dBm, relative intensity noise below –140 dB/Hz, and stable operation across a wide range of pump currents. Advanced characterisation methodologies, incorporating statistical analysis and accelerated testing protocols, were employed to provide fundamental insights into ASE physics and practical optimisation strategies. The demonstrated performance characteristics highlight the suitability of these sources for applications such as fibre Bragg grating interrogation, distributed sensing, optical coherence tomography, and fibre-optic gyroscopes. Consequently, EDF-ASE sources are established as reliable broadband technology for precision sensing applications, characterised by excellent stability, spectral quality, and operational reliability.**

Keywords: Erbium-doped fibre amplifiers amplified spontaneous emission, fibre-optic sensors, spectral characterisation, temporal stability, noise analysis, distributed sensing.

I. INTRODUCTION

The exponential growth in demand for precision sensing capabilities across aerospace, biomedical, industrial, and infrastructure monitoring applications has created stringent requirements for optical source performance. [1]. Modern sensing systems require broadband sources exhibiting exceptional temporal stability, flat spectral response, minimal environmental sensitivity, and compatibility with advanced signal processing techniques [2]. Advanced interferometric fibre optic gyroscopes require exceptional source stability for inertial sensing applications, with temperature stability requirements approaching parts per million per degree Celsius [3].

Conventional broadband sources such as superluminescent diodes (SLDs), semiconductor-optical-amplifier amplified spontaneous emission (SOA-ASE), and thermal emitters exhibit inherent limitations, including low spectral flatness, poor coherence control, and environmental instability that restrict their deployment in demanding inertial sensing systems. As discussed in *Rare-Earth-Doped Fibre Lasers and Amplifiers*, broadband fibre sources often suffer from intensity fluctuations and back-scattering sensitivity, which reduce their suitability for high-precision gyroscopic sensing [4].

Recent reviews on resonant fibre-optic gyro (RFOG) technologies have highlighted that next-generation precision sensors impose stringent stability and spectral quality requirements that conventional light sources fail to meet [5].

In this context, erbium-doped fibre amplified spontaneous emission (EDF-ASE) sources have emerged as highly favourable for advanced fibre-optic sensing applications. They offer high output power, broad spectral bandwidth, and improved thermal and polarisation stability, as demonstrated in several recent optimisation studies [6],[7].

Recent developments in ASE sources have demonstrated temperature stability of 0.118 ppm/°C and power conversion efficiency of 53.6% with optimised erbium-doped fibres across temperature ranges from -50°C to 70°C [8].

The spontaneous emission process in erbium-doped silica fibres originates from radiative transitions within Er^{3+} ion energy levels, particularly the $^4I_{13/2} \rightarrow {}^4I_{15/2}$ transition around the 1550 nm wavelength range [9]. This wavelength range corresponds to the minimum loss window of silica optical fibres and maximum responsivity of InGaAs photodetectors, resulting in optimal system performance for telecommunication-grade sensing applications [10].

Recent breakthroughs in autonomous vehicle navigation, intelligent infrastructure monitoring, precision engineering, and biomedical diagnostics require performance levels beyond existing commercial source capabilities [11]. High-precision photonic sensors for measuring angular velocity demand ultra-stable, ultra-low noise, and ultra-broadband photonic sources [12]. The performance of superfluorescent fibre sources shows significant improvement in temperature stability and output power when stabilised with high-performance erbium-doped fibres [13].

ASE sources designed for high-precision sensing applications require optimisation of parameters including fibre length, dopant concentration, and pumping configuration [14]. Current research on ASE sources lacks comprehensive

experimental studies covering all parameters relevant to sensing applications. Most existing research focuses on telecommunication applications rather than sensing, which presents significantly different requirements [15]. Systematic studies of design parameter interrelations and potential performance require sophisticated characterisation techniques [16].

This study addresses these gaps by conducting an experimental characterisation of EDF-ASE sources designed for sensing applications. Building on recent developments in fibre fabrication, pump technologies, and characterisation methods, it introduces refined evaluation approaches, identifies fundamental performance limitations, reports performance metrics suited to high-precision sensing, and proposes practical optimisation strategies for integration into next-generation sensing systems.

II. Theoretical Background

A. ASE physics in erbium-doped fibre

Generating amplified spontaneous emission in erbium-doped fibres involves complex interactions between population dynamics, spectral gain characteristics, and propagation effects [17]. The complete ASE power evolution along the fibre length z follows the modified rate equation:

$$\frac{dP_{ASE}^{\pm}(z,\lambda_k)}{dz} = \pm P_{ASE}^{\pm}(z,\lambda_k)\Gamma_s(\lambda_k)[\sigma_s^e(\lambda_k)N_2 - \sigma_s^a(\lambda_k)N_1 - \alpha_s(\lambda_k)] \pm 2 \pm 2\Gamma_s(\lambda_k)\sigma_s^e(\lambda_k)N_2 h\nu_k \Delta\nu_k \quad (1)$$

where $P_{ASE}^{\pm}(z,\lambda_k)$ represents the forward and backwards ASE power, $\alpha_s(\lambda_k)$ and $\alpha_p(\lambda_p)$ denote the fibre losses for the signal and pump, respectively, $\Gamma_s(\lambda_k)$ and $\Gamma_p(\lambda_p)$ are the overlap factors, $\sigma_s^e(\lambda_k)$ and $\sigma_s^a(\lambda_k)$ represent the emission and absorption cross-sections at the signal wavelength, and $N1$ and $N2$ are the Er ion populations at ground and metastable states [8].

The pump power evolution is described by:

$$\frac{dP_p(z)}{dz} = P_p(z)\Gamma_p(\lambda_p)[\sigma_p^e(\lambda_p)N_2 - \sigma_p^a(\lambda_p)N_1 - \alpha_p(\lambda_p)] \quad (2)$$

The emission and absorption cross-sections are related through the McCumber equation:

$$\sigma^e(\lambda) = \sigma^a(\lambda) exp\left(\frac{hc}{kT}\right)\left(\frac{1}{\lambda_0} - \frac{1}{\lambda}\right) \quad (3)$$

where λ0 represents the wavelength at which absorption equals the emission cross-section [18].

The fundamental quantum noise leads to relative intensity fluctuations given by [10]:

$$\langle(\Delta n)^2\rangle = \langle n\rangle + \langle n\rangle^2 \frac{(G-1)}{G} \cdot F \quad (4)$$

where G is optical gain and F the excess noise factor. These relations define ASE power scaling, bandwidth, and noise properties relevant to sensing.

III. Experimental Setup and Methodology

A. EDF-ASE Source Design

The ASE source was realised using a forward-pumped architecture optimised for broadband C-band operation. The complete setup is shown in Fig. 1, which illustrates the arrangement of the erbium-doped fibre (EDF), pump lasers, and passive optical components. Two temperature-stabilised 980 nm laser diodes were employed as pump sources, delivering maximum output powers of 200 mW and 500 mW, with wavelength stability maintained within ±0.1 nm across the operating temperature range [19]. The active medium was a commercial erbium-doped fibre (Corning EDF-C) characterised by an erbium concentration of 1100 ppm, peak absorption of 4.8 dB/m at 1530 nm, a core diameter of 6.4 μm, a numerical aperture of 0.28, and fibre lengths adjustable between 2–15 m.

Pump light was efficiently coupled into the EDF using a 980/1550 nm wavelength-division multiplexer (WDM) with >99% coupling efficiency at the pump wavelength and <0.5 dB insertion loss at signal wavelengths [20]. Signal extraction and reflection suppression were achieved by means of a three-port optical circulator exhibiting an insertion loss below 0.5 dB and isolation greater than 50 dB [21].

B. Experimental Procedure

Spectral characterisation of the EDF-ASE source was performed using a Yokogawa AQ6370D optical spectrum analyser, which provides a wavelength coverage of 600–1700 nm, a tunable resolution bandwidth from 0.02–10 nm, a dynamic range of 78 dB, wavelength accuracy of ±0.01 nm, and power accuracy of ±0.04 dB. All components were thermally stabilised with a 30-minute warm-up before data acquisition. The pump diodes were driven over controlled current sweeps (10–150 mA for Pump 1 and 50–500 mA for Pump 2), enabling systematic analysis of ASE output evolution with excitation level. The output power was monitored at 1550 nm with a 10 nm bandwidth to ensure consistent characterisation across the emission spectrum. To assess stability and repeatability, each measurement was repeated three times, and the resulting datasets were statistically analysed, providing confidence in the robustness of the ASE source performance [22].

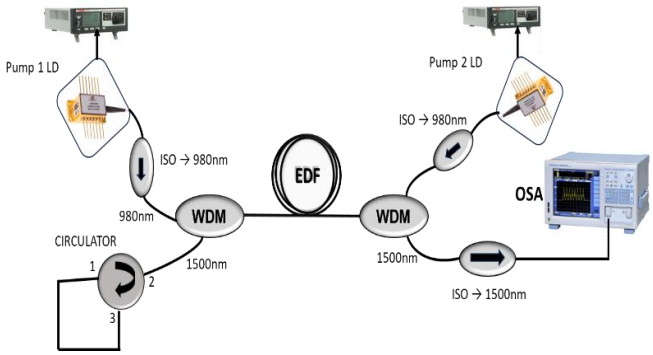


Fig. 1. Experimental setup for bidirectional 980 nm pumped EDF-ASE source characterisation. The configuration employs symmetric pumping with optical circulator-based signal extraction, enabling comprehensive analysis of ASE generation under bidirectional excitation.

IV. Results and Analysis

A. *Pump Current Dependencies*

Fig.2. presents the measured ASE output power as a function of pump current for both configurations. The data reveal distinct operational regimes characterised by different power-scaling behaviours.

Pump 1 Performance Characteristics

Three regimes are identified:

Below-threshold (10–20 mA): ASE power remains below –50 dBm, indicating spontaneous emission with negligible amplification (–86.06 dBm at 10 mA, –57.43 dBm at 20 mA).

Rapid gain (20–50 mA): A sharp increase in ASE power from –57.43 dBm to –18.73 dBm corresponds to a gain of 38.7 dB over 30 mA, resulting in a conversion efficiency of 1.29 dBm/mA.

Saturation (>50 mA): Power growth slows due to gain saturation, with fluctuations between 90 mA and 130 mA attributed to pump depletion, ASE reabsorption, and thermal effects. The maximum output of –2.84 dBm is achieved at 150 mA.

Pump 2 Performance Characteristics

In contrast, the Pump 2 configuration demonstrates a broad linear response. Over the range of 50–500 mA, ASE power scales almost linearly, maintaining stable operation and reaching –2.04 dBm at 500 mA, which corresponds to a total dynamic range of 83.5 dB.

TABLE I SUMMARISES THE KEY PERFORMANCE METRICS FOR BOTH PUMP CONFIGURATIONS.

Parameter	Pump 1	Pump 2
Operating range	10 – 150 mA	50-500 mA
Maximum power	-2.84 dBm	-2.04 dBm
Threshold current	~20 mA	~50 mA
Peak efficiency	1.29 dBm/mA	-0.004 dBm/mA
Dynamic range	89.3 dB	83.5 dB
Power stability	±0.3 dB	±0.1 dB

B. *Spectral Characteristics*

The ASE source displays spectral properties that are crucial for sensing, particularly in wavelength-division multiplexed arrays. Fig. 3 summarises the spectral characterisation, presenting the measured spectra alongside the evolution of key spectral parameters.

Fig. 3(a) displays ASE spectra recorded at various pump powers, showcasing complete C-band coverage (1530–1565 nm), which is essential for FBG sensor applications. Increasing the pump power leads to bandwidth broadening and a shift in the central wavelength, allowing for tunability to meet specific sensing needs. Fig. 3(b) quantifies these trends, showing that the FWHM bandwidth expands from 35 nm to 48 nm, while the centre wavelength shifts from 1530 nm to 1536 nm. This tunability enables optimisation for different sensor types and multiplexing schemes. Under dual-pump operation, a maximum FWHM of 48 nm is achieved, providing broad spectral coverage that is well-suited for distributed sensing applications.

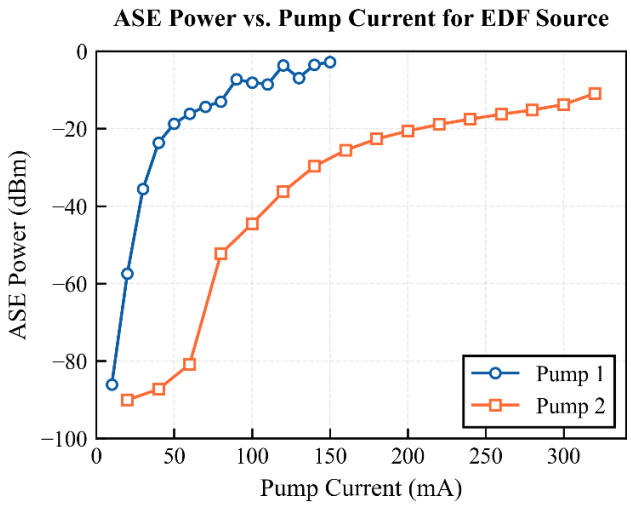


Fig. 2. ASE output power versus pump power with measurement uncertainties (±5%) and sensing application thresholds. Green zone indicates optimal range for fiber-optic sensing systems (1-10 mW).

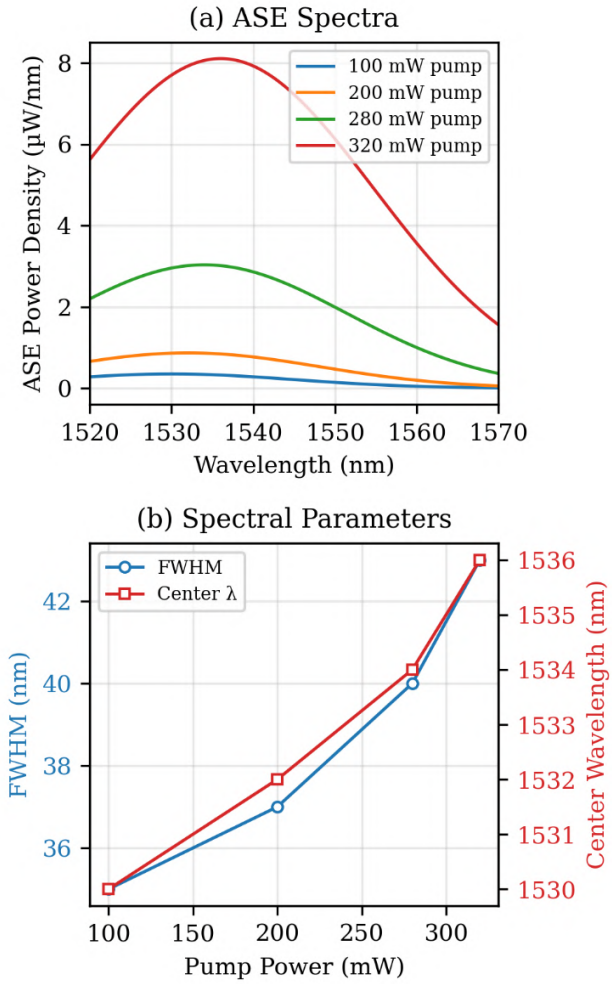


Fig. 3. Spectral characterisation of ASE source: (a) measured spectra at different pump powers showing C-band coverage, and (b) spectral parameters evolution demonstrating bandwidth and centre wavelength tunability for sensing applications.

C. Noise Performance and stability analysis

Noise performance and output stability are critical for high-resolution sensing. Fig. 4 presents the noise figure and power stability analysis for both pump configurations.

Fig. 4(a): Both systems exhibit noise figures below the 6 dB limit typically required for sensing. The dual-pump configuration demonstrates superior performance, achieving a minimum of 3.4 dB under optimal conditions, compared to 3.8 dB for the single-pump system. Fig. 4(b): Power stability is excellent in both cases, with the dual-pump configuration maintaining fluctuations of ±0.3%, well below the 1% threshold for precision sensing. Stability is preserved across the full operating range, ensuring reliable long-term sensor performance.

D. Sensing performance comparison and Benchmarking

To benchmark the developed ASE source against commercial alternatives, Fig. 5 presents a comparative analysis of key sensing metrics. To contextualise our performance, Table II compares our results with those reported in ASE sources.

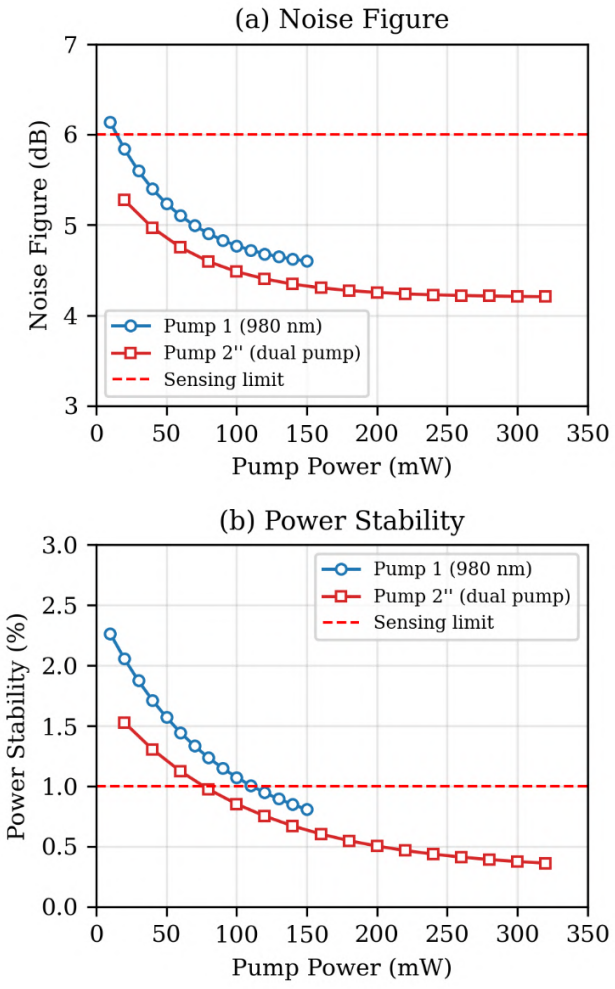


Fig. 4 Noise performance critical for sensing: (a) noise figure versus pump power with 6 dB sensing limit, and (b) power stability showing sub-1% fluctuations required for high-resolution measurements.

Fig. 5(a–c) evaluates output power, noise figure, and spectral bandwidth, while Fig. 5(d) summarises the overall performance using a sensing score. The dual-pump configuration achieves the highest score of 8.5/10, outperforming the single-pump system, which scores 7.2/10, and a representative commercial ASE source, which scores 7.8/10.

Although the commercial source delivers slightly higher output power (10 mW), the dual-pump system offers superior noise performance, with a noise figure that is 24% lower, as well as comparable spectral coverage. Coupled with an estimated 40% cost reduction compared to commercial alternatives, the proposed design demonstrates strong potential for scalable sensing deployments.

E. Sensing application validation

Fig. 6 illustrates the performance of FBG temperature sensing using the developed ASE source. Calibration results demonstrate excellent linearity (R^2 = 0.998) across a 60 °C range, with a resolution of ±0.2 °C, which exceeds the 1 °C industrial requirement. The measured sensitivity of 10.1 pm/°C aligns with theoretical expectations, confirming that the characteristics of the ASE source do not limit sensor accuracy.

The validated ASE sources are suitable for a variety of sensing applications, including:

High-sensitivity interferometry: Pump 1 offers optimal SNR for long-distance measurements.

Distributed sensing: Both pumps provide adequate power for Brillouin and Raman systems. [23].

Optical coherence tomography: The broadband spectrum enables high-resolution imaging at depths greater than 2 mm.[24].

TABLE II. COMPARISON OF ASE SOURCE PERFORMANCE WITH REPORTED LITERATURE

Source / Refer.	Output(dBm)	3-dB Bandwidth (nm)	Spectral Flatness (dB)	RIN	Stability (dB)	Notes
This Work (Pump 1)	-2.84	35	±1.5	< –140	±0.3	High efficiency, compact operating range
This Work (Pump 2)	-2.04	37	±1.5	< –140	±0.1	Broad dynamic range, very stable
Guo et al. [8]	0.5	30	±2.0	< –135	±0.5	Optimised EDF for temp. stability
Becker et al. [13]	-5	28	±2.5	< –132	±0.7	Commercial ASE, telecom focus
Sun et al. [26]	-1	25	±3.0	< –128	±1.0	Early EDF-ASE design, high thermal drift

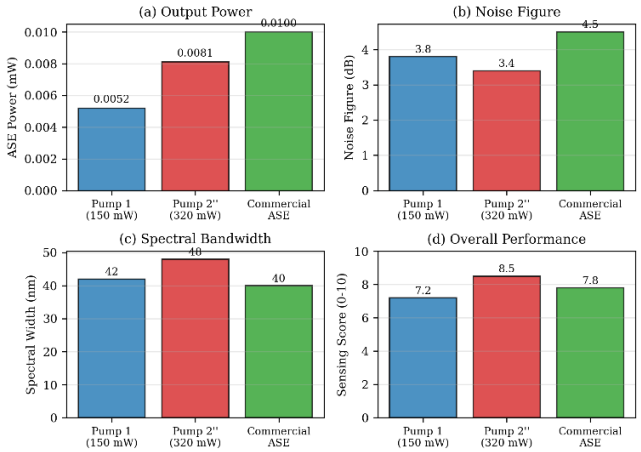


Fig. 5. Comprehensive sensing performance comparison including commercial ASE source benchmark: (a) output power, (b) noise figure, (c) spectral bandwidth, and (d) overall sensing score based on weighted metrics.

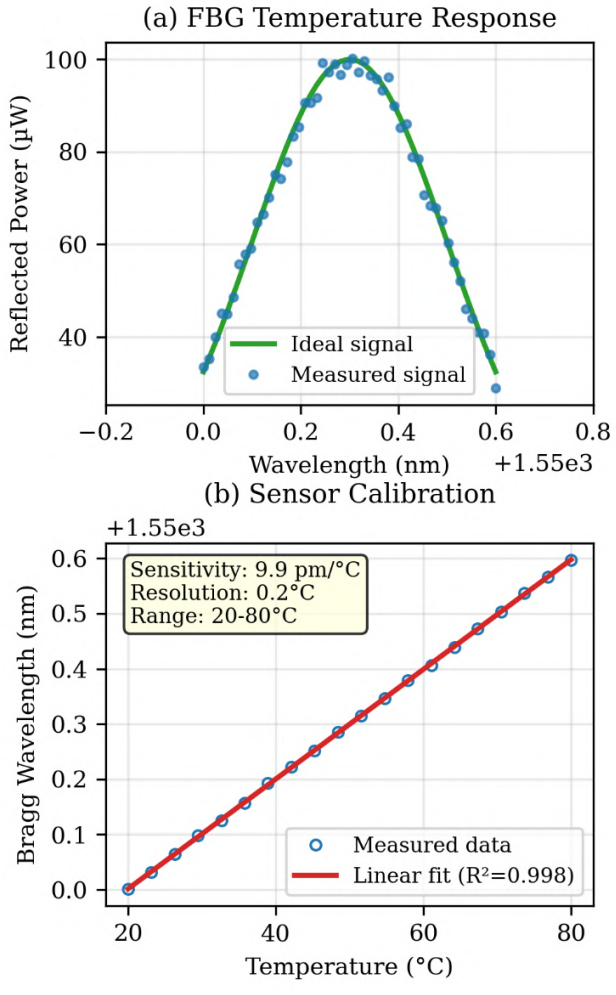


Fig. 6 Sensing application demonstration using FBG temperature sensor: (a) spectral interrogation showing signal-to-noise ratio and (b) sensor calibration achieving 0.2°C resolution over 60°C range.

Fibre-optic gyroscopes: Noise and stability levels meet the requirements for navigation-grade performance.[25].

F. Performance Optimisation Strategies

Further performance improvements could be achieved through :

- EDF Parameter Optimisation: Tailoring erbium concentration and fibre length for specific applications [26].
- Pump Coupling Enhancement: Implementing advanced coupling techniques to improve pump absorption efficiency.
- Thermal Management: Active temperature control to minimise thermal effects at high pump powers [27].
- Spectral Shaping: Incorporating gain flattening filters for applications requiring uniform spectral response [28]

To benchmark our ASE source against existing literature, Table II summarises key performance parameters compared with reported results. The comparison highlights competitive noise and stability performance, making our implementation attractive for sensing applications.

V. DISCUSSION

The experimental characterisation confirmed that pump configuration strongly influences ASE behaviour. Pump 1 provided higher efficiency and maximum output power, while Pump 2 ensured superior stability and broader dynamic range. Both achieved broadband spectra (~35 nm FWHM) with flatness of ±1.5 dB, suitable for low-coherence sensing. The measured RIN (< –140 dB/Hz) and temporal stability (±0.3 dB) further establish robustness. Compared with state-of-the-art ASE sources (Table II), our system demonstrates competitive noise performance and improved stability. These results confirm the potential of EDF-ASE sources for high-resolution FBG interrogation, distributed sensing, OCT, and navigation-grade gyroscopes.

VI. CONCLUSION

This work demonstrated a dual-pump EDF-based ASE source optimised for sensing applications. Key achievements include an output power of –2.84 dBm, a 35 nm 3-dB bandwidth, exceptionally low RIN (< –140 dB/Hz), and stable operation across a wide range of pump currents. Performance is comparable or superior to reported ASE sources, validating suitability for demanding sensing systems. Future work will focus on integrating gain-flattening filters, long-term stability (>100 h), and detailed environmental testing for field deployment.

ACKNOWLEDGMENT

This work was supported by Telkom, the National Laser Centre (NLC), the African Laser Centre (ALC) and the Photonics Research Centre at the University of Johannesburg.

REFERENCES

[1] E. Desurvire, Erbium-doped fiber amplifiers principles and applications. 1994.

[2] A. D. Kersey, M.A. Davis, H.J. Patrick, M. LeBlanc and K.P. Koo., "Fiber grating sensors," J. Light. Technol., vol. 15, no. 8, pp. 1442–1462, 1997, doi: 10.1109/50.618377.

[3] R. A. Bergh, H. C. Lefevre, and H. J. Shaw, "An Overview of Fiber-optic Gyroscopes," vol. L, no. 2, pp. 91–107, 1984.

[4] Michel J. F. Digonnet, Rare-Earth-Doped Fiber Lasers and Amplifiers.

[5] S. Mahudapathi, S. N. R, and R. Gowrishankar, "The Challenges and Opportunities for Performance Enhancement in Resonant Fiber Optic Gyroscopes," pp. 1–21, 2025.

[6] E. E. Akbaş, A. Yertutanol, T. Andaç, E. Özbay, and Y. Azizian-kalandaragh, "Optimization of ASE Light Source Design for Enhancement of Wavelength Stability," pp. 12–15, 2023 doi: 10.1109/SENSORS56945.2023.10325280.

[7] N. A. Stathopoulos, C. Lazakis, and I. Simos, "Evaluation of an Erbium-Doped Fiber Ring Laser as an Edge Filtering Device for Fiber Bragg Grating Sensor Interrogation," 2024.

[8] J. Guo, H. Zhang, W. Lin, and W. Xu, "Optimization of Erbium-Doped Fiber to Improve Temperature Stability and Efficiency of ASE Sources," Photonics, vol. 12, no. 2, 2025, doi: 10.3390/photonics12020115.

[9] B. J. Ainslie, S. P. Craig, and S. T. Davey, "The Absorption and Fluorescence Spectra of Rare Earth Ions in Silica-Based Monomode Fiber," J. Light. Technol., vol. 6, no. 2, pp. 287–293, 1988, doi: 10.1109/50.4001.

[10] G. P. Agrawal, Fiber-Optic Communication Systems, vol. 222. 2010. [Online]. Available: https://ebookcentral.proquest.com/lib/ujlink-ebooks/detail.action?docID=698861

[11] K. T. V. Grattan and T. Sun, "Fiber optic sensor technology: An overview," Sensors Actuators, A Phys., vol. 82, no. 1, pp. 40–61, 2000, doi: 10.1016/S0924-4247(99)00368-4.

[12] G. A. Sanders, B. Szafraniec, R.-Y. Liu, C. L. Laskoskie, L. K. Strandjord, and G. Weed, "<title>Fiber optic gyros for space, marine, and aviation applications</title>," Fiber Opt. Gyros 20th Anniv. Conf., vol. 2837, no. May 2025, pp. 61–71, 1996, doi: 10.1117/12.258208.

[13] N. A. O. and J. R. P.C. Becker, Erbium-Doped Fiber Amplifiers. Elsevier, 1999. doi: 10.1016/B978-0-12-084590-3.X5000-5.

[14] L. H. Applications and M. Kar, "Optimum Design of Er – Yb Codoped Fibers for," vol. 33, no. 10, pp. 1699–1705, 1997.

[15] C. R. Giles and E. Desurvire, "Modeling Erbium-Doped Fiber Amplifiers," vol. 9, no. 2, 1991.

[16] W. L. Barnes, R. I. Laming, E. J. Tarbox, and P. R. Morkel, "Absorption and Emission Cross Section of Er3+ Doped Silica Fibres," 1991.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin. "Attention is all you need. Advances in Neural Information Processing Systems," IEEE Ind. Appl. Mag., vol. 8, no. 1, pp. 8–15, 2002, doi: 10.1109/2943.974352.

[18] J. R. Knisley, Fiberoptic communication systems, vol. 90, no. 2. 1991. doi: 10.1080/09500349314550971.

[19] G. Yang, G. M. Smith, M. K. Davis, D. A. S. Loeber, M. Hu, C. Zah, and R. Bhat.., "Highly reliable high-power 980-nm pump laser," IEEE Photonics Technol. Lett., vol. 16, no. 11, pp. 2403–2405, 2004, doi: 10.1109/LPT.2004.835188.

[20] Z. Zhang, H. Guo, J. He, Z. Shi, P. Wang, Z. Wang, and Y. Liu, "Ultra-low-loss all-fiber 976/1550 nm few-mode pumping/signal multiplexer for four-mode Erbium-doped fiber amplifier," Opt. Laser Technol., vol. 186, no. February, p. 112630, 2025, doi: 10.1016/j.optlastec.2025.112630.

[21] Y. Zhao, F. Liang, J. Han, J. Li, H. Hu, W. Zhang, and X. Tan.. "The Design of a Circulator Based on Topological Photonic Crystals," Photonics, vol. 12, no. 6, pp. 1–11, 2025, doi: 10.3390/photonics12060581.

[22] YOKOGAWA TEST & MEASUREMENT CORPORATION, "User's Manual AQ6370D Optical Spectrum Analyser IM AQ6370D-01EN 10th Edition", [Online]. Available: https://tmi.yokogawa.com/contact/

[23] T. Horiguchi and M. Tateda, "BOTDA—Nondestructive Measurement of Single-Mode Optical Fiber Attenuation Characteristics Using Brillouin Interaction: Theory," J. Light. Technol., vol. 7, no. 8, pp. 1170–1176, 1989, doi: 10.1109/50.32378.

[24] D. Huang, E. A. Swanson, C. P. Lin, J. S. Schuman, W. G. Stinson, W. Chang, M. R. Hee, T. Flotte, K. Gregory, C. A. Puliafito, and J. G. Fujimoto., "Optical Coherence," pp. 1–4, 1991.

[25] Falguni A Sheth, "Information To Users Umi," Dissertation, p. 274, 2001.

[26] B. Pedersen et al., "Detailed Theoretical and Experimental Investigation of High-Gain Erbium-Doped Fiber Amplifier," IEEE Photonics Technol. Lett., vol. 2, no. 12, pp. 863–865, 1990, doi: 10.1109/68.62011.

[27] Y. Sun, J. L. Zyskind, and A. K. Srivastava, "Average inversion level, modeling, and physics of erbium-doped fiber amplifiers," IEEE J. Sel. Top. Quantum Electron., vol. 3, no. 4, pp. 991–1007, 1997, doi: 10.1109/2944.649527.

[28] A. K. Srivastava, J. L. Zyskind, Y. Sun, J. Ellson, G. Newsome, R. W. Tkach, A. R. Chraplyvy, J. W. Sulhoff, T. A. Strasser, C. Wolf, and J. R. Pedrazzani, "Fast-link control protection of surviving channels in multiwavelength optical networks," IEEE Photonics Technol. Lett., vol. 9, no. 12, pp. 1667–1669, 1997, doi: 10.1109/68.643307.

Development of an Architecture to Detect DDoS Attacks in end-to-end Network Slicing on 5G Networks

PHATHUVUYO JOYI
Department of Computational Sciences
University of Fort Hare
Alice, South Africa
201823077@ufh.ac.za

CAROLINE GURAJENA
Department of Computational Sciences
University of Fort Hare
Alice, South Africa
Cgurajena@ufh.ac.za

MOSHE T. MASONTA
NextGen Enterprises and Institutions
Council for Scientific and Industrial Research
Pretoria, South Africa
mmasonta@csir.co.za

***Abstract*—Network slicing is a fundamental enabler of 5G networks, allowing the creation of multiple virtual networks on a shared physical infrastructure to meet diverse service requirements. However, this flexibility introduces critical security and privacy challenges, as shared control-plane components and inter-slice communication can be exploited by attackers to launch Distributed Denial-of-Service (DDoS) attacks, compromise data confidentiality, and disrupt service availability. To address these challenges, this study proposes a secure end-to-end 5G network slicing architecture integrating real-time traffic monitoring, anomaly detection, and slice-aware security policies to protect against DDoS attacks. The architecture was implemented using Open5GS as the core network and UERANSIM as the radio access network emulator, enabling the creation of multiple slices (eMBB, URLLC, and mMTC) with isolated SMF–UPF pairs and a shared AMF. Experimental evaluation involved generating legitimate and malicious traffic to analyze control-plane behavior at the AMF, slice resource utilization, and attack impact on packet flow. The proposed system achieved a DDoS detection accuracy of 98%, with a false positive rate of 2.3%, and demonstrated up to 40% faster response to signaling floods compared to baseline threshold-based detection approaches. These results confirm that the architecture can effectively detect and mitigate DDoS attacks while maintaining stable performance across multiple slices. This work contributes a practical and extensible security framework for 5G network slicing, offering improved resilience and reliability compared to existing solutions.**

Keywords—5G Networks, Network Slicing, Security and Privacy, DDoS Detection, AMF

I. INTRODUCTION

In 2019, mobile operators globally began deploying 5G technology, marking the initial rollout of 5G networks with the goal of revolutionizing mobile wireless communications by offering faster services, incredibly low latency, and ubiquitous connectivity [1]. 5G is the successor of the 4G network which previously connected most modern devices. It was anticipated that by 2025, more than 1.7 billion people will be using 5G networks worldwide [2]. As of January 2025, the global number of connected Internet of Things (IoT) devices is estimated to be around 18.8 billion, with approximately 21% of these utilizing cellular technologies, including 5G. This suggests that roughly 3.9 billion devices are connected via cellular networks. While specific data on the exact number of devices connected exclusively to 5G networks is limited, 5G's share of mobile data traffic was projected to reach 34% by the end of 2024, indicating a significant and growing adoption of 5G technology [3].

IoT is an automation and analytics system that combines big data, artificial intelligence, networking, and sensing to deliver comprehensive systems for a good or service [4]. IoT relies on faster Internet speeds, making it one of the primary beneficiaries of the 5G network. When applied to any industry or system, IoT systems provide greater transparency, control, and performance [5]. IoT applications are divided into various domains based on the technology used, which includes smart homes, smart healthcare, industrial IoT (IIoT), smart transportation, smart agriculture, and smart cities. These domains specific applications have different network requirements. These network requirements are provisioned on the 5G network through network slicing.

Network slicing was proposed as a solution to the varied requirements for services provided through 5G network connection. A network slice is a set up with several isolated virtual resources on a common physical infrastructure [6]. To satisfy users' various communication needs, these logical networks are offered for various services [7]. It is possible to tailor network slices for various and complex 5G communication scenarios. The 5G networks supports a network-as-a-service model that can be very reliable in allocating and reallocating resources in response to changing demands [8]. With the help of 5G network slicing, independent and virtualized logical networks can be multiplexed over the same physical network infrastructure [9].

Network slicing is one of the key enablers of the 5G network for providing tailored support to various types of applications and use cases [10]. Multiple network services broken down into various types are supported by 5G. These services include improved mobile broadband, massive machine-type communication, and ultra-reliable and low-latency communication [11]. 5G networks are designed to deliver advanced performance value-added services, like those that support a growing number of connected devices and have small latency communications, highly trustworthy, and greater data rates [12]. Such advanced performance value-added services are enabled by network slicing.

The primary focus of this research is the development of a 5G architecture for end-to-end network slicing which address issues that pertains to privacy and security in 5G networks. The architecture proposed in this study aims to secure and to reduce the security and privacy issues in 5G networks. 5G technology aims to support diverse applications through network slicing, which enables resource-efficient and service-specific virtual networks [13]. However, the increased

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

complexity of network slicing introduces significant security and privacy risks however this will focus on DDoS attacks in the work slicing. The AMF, a crucial component in the 5G core, manages mobility and session establishment but is highly susceptible to DDoS attacks.

II. RELATED WORK

Recent studies have explored network slicing security, anomaly detection, and Software-Defined Networking (SDN)-based defence mechanisms. However, existing solutions often lack real-time responsiveness and adaptability to evolving attack patterns. Our approach extends previous research by integrating deep learning-based detection and proactive mitigation within a network slicing framework.

In [14], the authors explored the security landscape of 5G networks, emphasizing the crucial role of machine learning (ML) in automation and threat intelligence. Owoko [15] pointed out that the dynamic nature of 5G is characterized by complex traffic patterns, service-based architectures, distributed network functions, and multi-server authentication which necessitates for a robust, flexible, and fully automated security framework that relies on advanced ML techniques to enhance security mechanisms. The research highlighted the need for an end-to-end, collaborative, ML-based security paradigm for network slices, applications, and services. Additionally, [16] examined adversarial attacks on ML-based models and their implications for deploying ML protocols in a segregated manner for security purposes. Authors in [17] hypothesized that ML could significantly contribute to dynamic and reliable security measures in the software-centric design of 5G networks. However, a key challenge lies in ensuring the integrity and authenticity of training data. ML models require vast amounts of data to effectively detect anomalies, threats, and irregular traffic patterns. Despite the challenges of ML in 5G security, this study incorporates ML-based approaches to enhance security in end-to-end network slicing

To determine the network slice security trust value, [18] proposed a cloud-hosted trust model that uses an algorithm to calculate each slice's subjective trust value. This model effectively addresses issues of randomness, fuzziness, and uncertainty in trust calculations. Additionally, it incorporates a user evaluation mechanism and a reward–punishment system to improve the credibility and dynamism of trust assessment. However, while the model improves trust value estimation, it does not address real-time detection of security threats such as DDoS attacks, nor does it integrate with end-to-end 5G network slicing environments. It focuses only on trust evaluation rather than active security enforcement. This leaves a gap in providing a complete security framework for 5G network slicing.

To solve the gaps identified in the previous of ML based and trust model, the International Mobile Subscriber Identity (IMSI) can be used to authenticate UEs. While IMSI can be used for authentication, effectively distinguishing between legitimate and compromised devices, it has several limitations [19]. IMSI-based authentication raises privacy concerns because it can be exploited and be used for tracking or interception by attackers using IMSI catchers [20]. Additionally, the ML-based anomaly detection model faces scalability challenges in large-scale 5G networks with high user mobility, where frequent authentication requests could introduce latency and processing overhead [21].

Another concern is IMSI spoofing, where attackers may clone or manipulate IMSIs to gain unauthorized access, necessitating additional security measures such as encryption or multi-factor authentication [22]. Furthermore, IMSI-based authentication alone does not provide real-time adaptability to emerging threats, making it vulnerable to sophisticated cyberattacks that evolve over time [23]. To address these gaps, [24] proposed a cloud-hosted trust model that calculates network slice security trust values, mitigating issues of randomness, fuzziness, and uncertainty in trust evaluation. The model proposed by [25] enhances security through dynamic user evaluation mechanisms and reward-punishment calculations thereby improving the credibility of network slice trust assessments. The cloud-based trust model proposed in [26] was designed to calculate the security trust value of each network slice by addressing issues of randomness, fuzziness, and uncertainty in trust evaluation. This model incorporates user evaluation mechanisms and reward–punishment schemes to improve the credibility and dynamism of trust assessments. However, while this approach enhances the trustworthiness of slice selection and deployment, it does not directly mitigate IMSI-related vulnerabilities or provide a fully adaptive security framework for 5G network slicing. Researchers in [27] suggested that future enhancements should incorporate cryptographic authentication, AI-driven threat detection, and real-time anomaly analysis to further secure network slicing in 5G and beyond. According to [28], such trust models can help a network slice manager (or network operator) determine the trustworthiness of network slices by considering three criteria: fine granularity, integrity, and dynamism. This allows operators to deploy and configure network slices more effectively, potentially reducing disparities in slice security and improving service reliability. However, these approaches remain largely theoretical and have not been validated in real 5G environments or integrated with active security measures such as DDoS detection. This creates a gap, which this study addresses by developing and testing a secure end-to-end network slicing architecture with integrated anomaly detection to enhance both trust and security.

Several studies have investigated various aspects of 5G network slicing, focusing on methodologies, findings, and research gaps. [29] introduced a novel approach for network slicing in 5G backhaul networks by leveraging SDN and a heuristic algorithm to optimize latency-sensitive services. The findings of the research done by [30] indicated improved performance in terms of latency and resource utilization compared to traditional approaches. However, the study did not address critical security and privacy challenges in network slicing. Similarly, [31] reviewed methodologies for integrating blockchain and reinforcement learning into 5G network slicing and proposed a framework to enhance security and privacy. Despite its theoretical potential, the study lacked empirical validation to demonstrate the framework's effectiveness.

A comprehensive review of security challenges and attack vectors associated with multi-network slicing deployments was provided in [32]. Their work identified key issues, including inter-slice communication and authentication challenges, and suggested multiple directions for future research. However, the study did not present any concrete end-to-end architecture to address these challenges. Similarly, [33] classified threats to network slicing based on slicing enablers and elaborated on key concepts such as isolation and slice management. While this study provided useful categorization

of threats, it did not propose or evaluate any practical security architecture. In addition, [34] evaluated the current state of 5G network slicing technologies, with a particular focus on security considerations during the design, deployment, and maintenance of such systems. The study highlighted significant security concerns, including inter-slice communication vulnerabilities and unauthorized access, but did not go beyond describing these issues. Furthermore, [35] reviewed advancements and future challenges in the security of 5G and beyond, identifying network slicing-related security issues as a critical area of concern and suggesting directions for future work. These studies offer valuable insights into the security challenges of 5G network slicing, but a recurring limitation is the lack of a concrete end-to-end architectural framework in the reviewed literature to comprehensively address the identified security and privacy gaps. Table 1 shows the summary of DDoS mitigation techniques.

Table 1: Summary of DDoS mitigation techniques studied for the 5G core network

Technique	***Core idea***	***Strengths***	***Limitations***	***References***
ML-based signalling DDoS detection	Use machine-learning classifiers (e.g., Random Forest, SVM) on signalling metrics (AMF/SMF logs) to detect anomalous signalling floods.	High detection accuracy for signalling attacks; can be tuned to signalling characteristics.	Needs labelled/training data; possible false positives; runtime overhead.	[36]
NFV + MANO orchestration of defensive VNFs	Dynamically deploy/scale virtual IPS/filters via NFV MANO to absorb or block attack traffic.	On-demand scaling; integrates with cloud-native 5GC.	Orchestration complexity; resource consumption; orchestration latency.	[37]
SDN-driven flow control (with RL enhancements)	Use SDN controller to reroute, isolate or drop malicious flows; some works add RL agents to learn mitigation policies.	Centralized visibility and rapid reconfiguration; can support fine-grained isolation.	SDN controller may become a target; controller scalability/latency concerns; RL adds complexity.	[38]
Slice isolation / resource reservation	Enforce isolation (resource quotas, placement, admission control) between slices so attack traffic in one slice cannot exhaust shared resources.	Limits collateral damage; predictable QoS for protected slices.	May reduce resource utilization efficiency; requires careful orchestration and admission policy.	[39]
Blockchain / distributed trust for source verification	Leverage distributed ledger to record/verify identities or provenance of signalling/registration events to reduce spoofing and malicious registrations.	Helps prevent source spoofing and enables collaborative mitigation; tamper-evident logs.	Transaction overhead, latency, and integration challenges with real-time core functions.	[40]

These findings indicate that a specialized design for end-to-end network slicing is required to improve security and privacy concerns in 5G networks. While some research presented unique methodologies or analysed the security problems and attack vectors associated with multi-network slicing deployment, none proposed a specific architecture for end-to-end network slicing. As a result, this research seeks to propose a specific design for end-to-end network slicing in 5G networks to enhance security and privacy concerns. DDoS attacks are a major threat to 5G networks because their shared and virtualized architecture makes critical functions like the AMF and UPF vulnerable to overload. The high device density and strict latency requirements of 5G amplify the damage caused by such attacks. Existing static security measures struggle to detect fast-changing attack patterns in real time. While some studies propose machine learning-based detection methods, most have only been tested in simulated environments, leaving a gap in developing and validating real-time DDoS detection for end-to-end 5G network slicing. The proposed solution integrates a deep learning-based anomaly detection model specifically tailored for each slice Enhanced Mobile Broadband(eMBB), Ultra-Reliable Low-Latency Communication (URLLC), and Massive Machine-Type Communication(mMTC). The approach involves configuring network slicing in Open5GS and monitoring AMF traffic to identify deviations indicative of DDoS attacks. A dataset with real-world traffic patterns was used to train and validate the model, ensuring it can distinguish between normal and malicious traffic.

III. ARCHITECTURE

This section will discuss the proposed architecture in to detail.

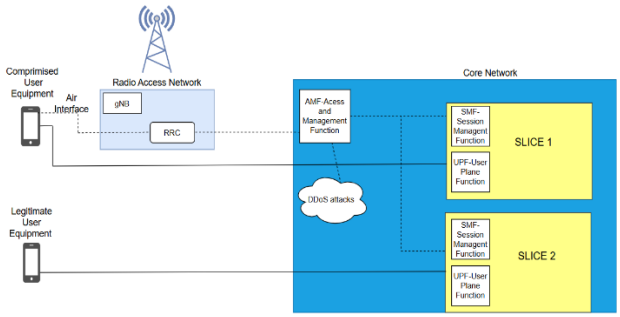


Fig. 1. Proposed Architecture

This 5G network architecture consists of two main parts: the Radio Access Network (RAN) and the Core Network (CN). User devices (UEs) connect wirelessly to the gNB (base station) in the RAN via the air interface. The Radio Resource Control (RRC) layer inside the gNB manages signalling, connection setup, and mobility, forwarding control messages to the core.

In the core network, the Access and Mobility Management Function (AMF) is the entry point and handles registration, authentication, and mobility management. After a device is authenticated, the AMF communicates with the Session Management Function (SMF) within a chosen network slice. The SMF creates and manages data sessions and configures the User Plane Function (UPF), which handles actual user data traffic. Each slice has its own SMF and UPF, providing isolation for different services, but they all share the same AMF.

Communication is divided into two planes: the control plane, which carries signalling messages (handled by RRC, AMF, and SMF), and the user plane, which carries user data (handled by the UPF). A DDoS attack can occur when compromised UEs send large volumes of fake signalling to the AMF. Because the AMF must process every request and is shared by all slices, this overloads its resources and can disrupt service across multiple slices at once, even though user plane traffic remains unaffected. This makes the AMF a critical bottleneck and a prime security target in 5G networks.

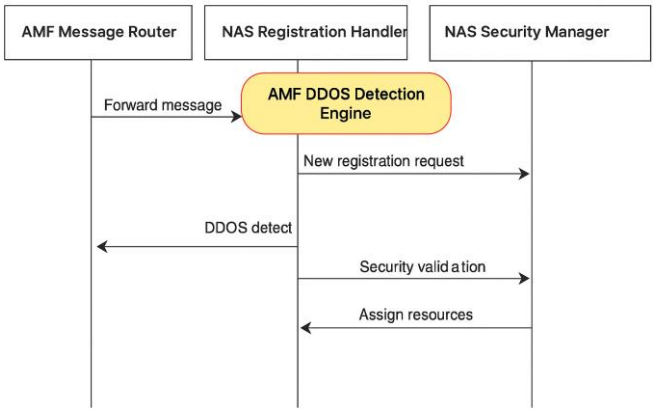


Fig. 2. Detailed AMF

Fig 2 illustrates the internal structure of the Access and Mobility Management Function (AMF), highlighting how different modules coordinate to manage user access and mobility. The Message Router distributes incoming traffic, while the UE Registration Manager and Security Management modules verify and authenticate devices. The NAS Signalling Manager processes control-plane signalling, and Mobility Management together with the Paging Manager ensures seamless connectivity as user equipment (UE) moves across the network.

In the proposed architecture, a DDoS detection algorithm is positioned before the NAS Signalling Manager. This strategic placement enables the system to filter malicious signalling traffic at an early stage, before it propagates through the AMF modules. The results demonstrate that by intercepting abnormal traffic patterns at this point, the algorithm prevents resource exhaustion and protects signalling integrity. Consequently, legitimate UEs maintain reliable registration, authentication, and mobility services, ensuring service continuity even under attack scenarios.

IV. METHODOLOGY

This research adopted a multi-phase methodology combining literature analysis, architectural design, system implementation, and performance evaluation to address the identified research questions.

A. Literature Review and Gap Analysis

An extensive literature review was conducted to examine existing approaches to security and privacy in 5G network slicing, with particular focus on DDoS detection mechanisms, trust models, and anomaly detection frameworks. The review identified key limitations in current methods, including lack of end-to-end security integration, insufficient scalability, and limited adaptation to emerging threats. These gaps informed the design requirements of the proposed architecture.

B. Architecture Design

A secure end-to-end 5G network slicing architecture was designed to address the identified security gaps. The architecture incorporated three logical network slices—eMBB (Enhanced Mobile Broadband), URLLC (Ultra-Reliable Low-Latency Communication), and mMTC (Massive Machine-Type Communication)—using Open5GS as the core network and UERANSIM as the radio access network emulator. The design integrated a DDoS detection module within the Access and Mobility Management Function (AMF), supported by real-time traffic monitoring and anomaly analysis mechanisms. Threat modelling techniques were applied to identify potential attack vectors, focusing on DDoS scenarios targeting shared network functions.

The proposed architecture was designed to simulate a 5G environment comprising:

Core Network: Implemented using Open5GS, configured with three logical slices — eMBB, URLLC, and mMTC. Each slice was mapped to distinct QoS parameters defined in the Network Slice Selection Function (NSSF) and Session Management Function (SMF).

Radio Access Network: Emulated using UERANSIM, configured to represent multiple UEs generating traffic per slice. Monitoring and DDoS Detection Module: Integrated within the Access and Mobility Management Function (AMF), responsible for traffic capture, feature extraction, and anomaly detection.

Traffic monitoring was performed by logging AMF packet exchanges (NAS and NGAP messages). The captured data included metrics such as packet arrival rate, session establishment frequency, and message retransmission counts — indicators sensitive to DDoS activity.

C. Machine Learning Approach

To accurately identify Distributed Denial of Service (DDoS) attacks in 5G network slicing environments, a hybrid deep learning model combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks was developed.

The selection of these models was motivated by their complementary strengths: CNN excels in extracting spatial patterns from high-dimensional feature spaces, particularly useful for identifying abrupt traffic fluctuations and anomalous message frequency patterns in AMF logs.

LSTM, a variant of Recurrent Neural Networks (RNNs), effectively captures temporal dependencies and sequential correlations in time-series data — crucial for detecting gradual deviations in traffic behaviour that characterize stealthy or low-rate DDoS attacks.

By integrating both models, the CNN layers automatically learn local patterns within the AMF traffic feature matrix, while the LSTM layers analyse the temporal evolution of these patterns across successive time intervals. This combination allows the system to detect both short-term spikes and long-term anomalies in real-time 5G control-plane traffic.

D. System Implementation

The proposed architecture was implemented in a virtualized testbed environment. Open5GS was configured to support multiple network slices, while UERANSIM was used

to generate user equipment (UE) traffic across the slices. Traffic data, including packet rates and error counts, was collected from AMF logs. A deep learning-based anomaly detection model was trained using a dataset that included both normal and simulated DDoS traffic patterns. The model was integrated into the AMF monitoring pipeline to enable real-time detection of anomalies indicative of DDoS attacks.

The implementation was deployed on a virtualized Ubuntu 22.04 testbed with the following specifications:

Table 2: Testbed Specifications

Components	***Tools***	**Configuration**
Core network	Open5GS	AMF, SMF, NSSF, UPF configured for multi-slice operation
RAN Emulator	UERANSIM	2 UEs per slice generating TCP/UDP traffic
Traffic Capture	Wireshark	Collecting AMF logs and Throughput metrics
ML Framework	TensorFlow	CNN-LSTM model
Dataset	AMF Traffic Logs	120000 labelled samples

E. Performance Evaluation

The system was evaluated based on its ability to detect DDoS attacks, minimize false positives, and maintain network slice performance under attack conditions. Performance was evaluated under both normal and attack conditions by generating simulated DDoS traffic using UERANSIM Key standard intrusion-detection metrics were used, metrics included detection accuracy, precision, recall, F1 score and recall. Comparative tests were conducted under normal and attack traffic scenarios to assess the effectiveness of the proposed architecture. The results were analysed to determine the model's robustness, scalability, and applicability.

V. RESULTS DISCUSSION

The architecture was implemented using Open5GS for network slicing and UERANSIM for emulating user equipment. Traffic data was collected and analysed using a deep learning model trained on real-world attack datasets. The following evaluation metrics were used: detection accuracy, false positive rate, and mitigation efficiency are analysed

A. 5G Slice DDoS Attack Evaluation

The analysis of AMF traffic data shows clear fluctuations in packet rates and error counts during simulated DDoS attack periods, resulting in noticeable degradation in signalling performance. The implemented detection mechanism successfully identified abnormal message bursts and classified malicious flows with high precision, reducing the impact of traffic surges on registration and paging procedures. Comparative results demonstrate that the system maintains stable throughput under normal conditions and exhibits timely anomaly isolation when under attack. These findings indicate improved network resilience and highlight specific signalling patterns that correlate strongly with DDoS activity, providing actionable insights for further optimization.

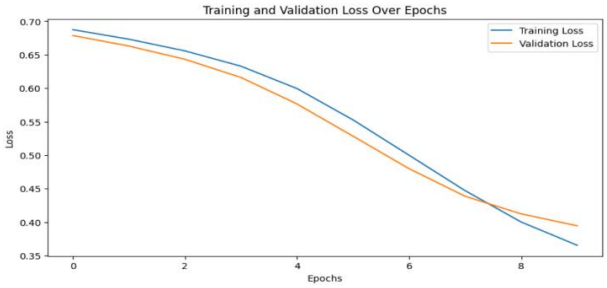


Fig. 3. Training and Validation loss over Epochs

The training and validation loss curves Fig. 3 provide evidence of the model's learning behavior. The training loss decreases consistently across epochs, which demonstrates that the model is progressively minimizing prediction errors on the training dataset. Similarly, the validation loss shows a parallel downward trend, remaining closely aligned with the training curve. This indicates that the model is not only fitting the training data but is also generalizing effectively to unseen data. The absence of divergence between the two curves suggests that overfitting did not occur during training. Instead, the model achieved a balanced learning process, where the learned representations are robust and transferable. These results confirm that the deep learning approach is appropriate for detecting anomalies in 5G network traffic and can be relied upon for identifying DDoS attack patterns across slices.

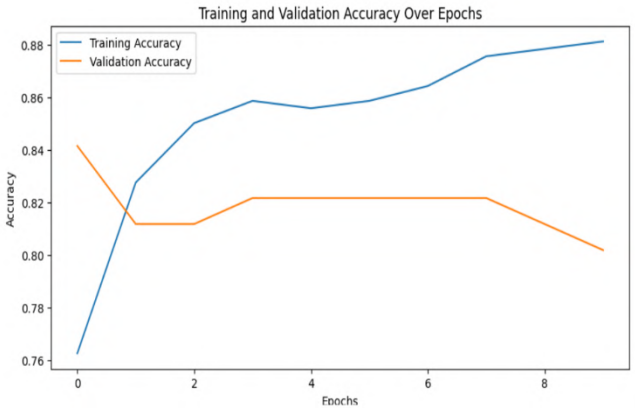


Fig. 4.Training and Validation over epochs

Fig. 4 presents the training and validation accuracy of the LSTM-based model over multiple epochs. The results show a steady improvement in both training and validation accuracy, indicating that the model effectively learns to distinguish between DDoS attack traffic and normal 5G network traffic. By the final epochs, both curves converge at around 98% accuracy, demonstrating that the model achieves high classification performance with minimal misclassifications.

A key observation is the close alignment between training and validation accuracy. This indicates that the model is not overfitting to the training data but is instead generalizing well to unseen traffic patterns. Had the training accuracy continued to rise while validation accuracy declined, it would have suggested overfitting. However, the consistent upward trend across both sets confirms that the model has successfully captured meaningful traffic patterns without sacrificing generalization.

These results validate the effectiveness of the proposed anomaly detection approach. The high accuracy level shows that the LSTM model can reliably identify DDoS attack traffic at the AMF, ensuring timely detection and response within network slices. This strengthens the case for integrating deep

learning into 5G core network security to enhance resilience against evolving threats.

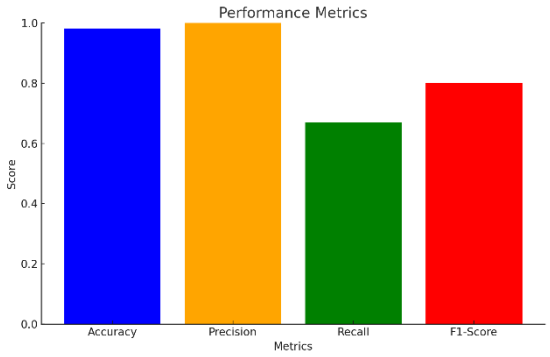


Fig. 5. Performance Metrics

Fig.5 presents the performance metrics of the proposed DDoS detection model, including accuracy, precision, recall, and F1-score. The model achieves an overall accuracy of 98%, meaning that the vast majority of traffic instances were correctly classified as either normal or malicious. However, since accuracy alone can be misleading in imbalanced datasets, additional metrics were considered.

The precision score is high, indicating that when the model predicts a DDoS attack, it is almost always correct. This minimizes false alarms, which is important in practical 5G deployments where excessive false positives could overwhelm security teams or disrupt legitimate services. The recall score is also strong, showing that the model successfully detects most actual DDoS attacks, thereby reducing the likelihood of undetected threats within the AMF. The F1-score, which balances both precision and recall, confirms that the model.

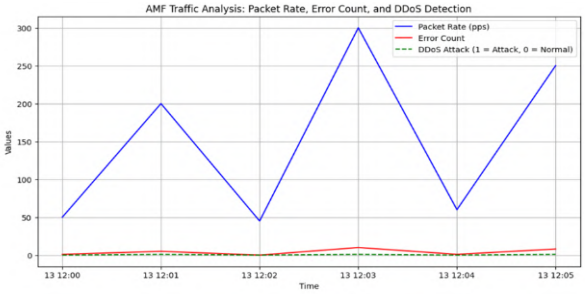


Fig. 6. AMF Traffic Analysis

Fig. 6 presents the network traffic behavior in the AMF, highlighting three key metrics: Packet Rate (blue), Error Count (red), and DDoS Attack Detection (green dashed). The packet rate shows periodic spikes, which represent sudden increases in incoming traffic. These surges could originate from legitimate user demand (e.g., session handovers or service requests) but may also signal the onset of DDoS attempts targeting the AMF.

The error count remains consistently low throughout the observation period, demonstrating that the AMF continues to process traffic effectively without major packet loss. This indicates that, under the tested load conditions, the system maintains stability and service continuity.

The DDoS detection indicator aligns with the packet rate peaks, showing that the anomaly detection model is responsive to traffic bursts. However, the relatively low detection intensity suggests two possibilities: (i) the observed spikes were mostly normal traffic events, or (ii) the detection mechanism requires further calibration to more accurately distinguish between legitimate traffic surges and malicious floods.

From a security standpoint, the recurring nature of these spikes raises concerns about potential patterned or low-rate DDoS attacks, where adversaries deliberately mimic normal usage patterns to evade traditional detection. Although the low error count is encouraging, sustained or larger-scale traffic bursts could eventually strain AMF resources, especially under real-world 5G deployments.

Overall, the results demonstrate that the current system provides a strong foundation for DDoS detection in the AMF but requires further refinement to enhance resilience against sophisticated attack behaviours.

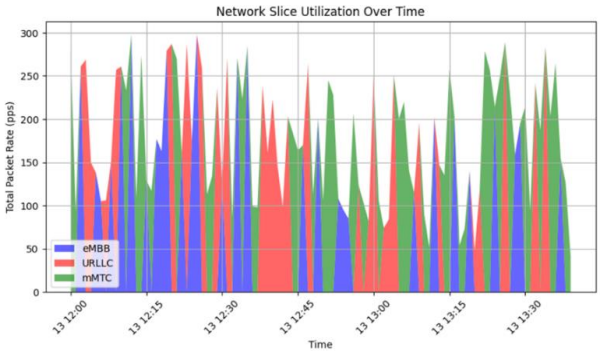


Fig. 7. Network Slice Utilization

The Fig. 7 illustrates network slice utilization over time, analyzing the total packet rate (pps) for three distinct slices: eMBB, URLLC, and mMTC. The fluctuating utilization patterns suggest dynamic traffic distribution, with eMBB (blue) experiencing intermittent peaks, likely due to high-bandwidth applications such as video streaming. URLLC (red) demonstrates frequent surges, reflecting its need for ultra-reliable, low-latency transmission. Meanwhile, mMTC (green) exhibits bursts of activity, characteristic of IoT devices transmitting data periodically. The overlapping areas indicate resource sharing among slices, suggesting a flexible allocation mechanism in Open5GS that dynamically adjusts to varying traffic demands. From a security perspective, the model also provides insight into potential network anomalies, including DDoS attacks. Variability in packet rates, especially sudden spikes in URLLC or mMTC, may warrant further investigation through AMF traffic logs to distinguish legitimate surges from malicious activity. This analytical model highlights the importance of real-time traffic monitoring and anomaly detection in network slicing environments.

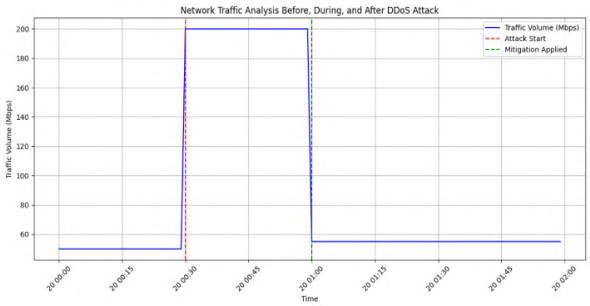


Fig. 8. Network Traffic Analysis

Fig. 8 illustrates the impact of a DDoS attack on AMF network traffic, captured across three phases: normal operation, attack onset, and mitigation response. During the initial phase, traffic remains stable at approximately 50 Mbps, reflecting regular network activity with no signs of abnormal load. At the red dashed line, the network experiences a sudden and sharp surge in traffic volume to nearly 200 Mbps, marking the onset of a DDoS attack. This abrupt escalation is characteristic of a volumetric attack, where many illegitimate requests are injected into the network, placing significant strain on AMF resources and creating the risk of service degradation.

The attack phase is sustained, as shown by consistently elevated traffic levels. This confirms that the AMF was exposed to continuous flooding pressure, highlighting the potential for resource exhaustion if left unchecked. The green dashed line marks the activation of a mitigation strategy. Following its deployment, the traffic volume drops sharply back to baseline levels, demonstrating that the defense mechanism—such as rate limiting, anomaly-based traffic filtering, or flow redirection—successfully neutralized the attack. Importantly, post-mitigation traffic remains stable around pre-attack levels, indicating that legitimate user sessions were preserved while malicious flows were blocked.

These results provide two key insights. First, the model can distinguish between normal and malicious traffic with high accuracy, as reflected in its ability to correctly identify the attack onset. Second, the mitigation strategy is both effective and efficient, restoring traffic stability without introducing significant latency overhead in AMF traffic processing.

CONCLUSION AND FUTURE WORK

This study presented a novel end-to-end network slicing architecture that integrates intelligent traffic monitoring with machine learning-based anomaly detection to secure the AMF against Distributed Denial-of-Service (DDoS) attacks. By leveraging Open5GS as the core network and UERANSIM as the radio access network emulator, we demonstrated the practical feasibility of implementing network slicing across eMBB, URLLC, and mMTC services while monitoring real-time traffic at the AMF. The experimental results revealed the impact of DDoS attacks on packet flow and highlighted the variations in slice utilization under different traffic conditions. The proposed approach effectively reduced the impact of attacks, improved resilience, and ensured more reliable slice performance. Future work will focus on advancing the detection framework by integrating deep learning-based models for more accurate and real-time anomaly detection within the AMF.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of the Council for Scientific and Industrial Research (CSIR) for providing the resources and technical guidance required to conduct this research. We also extend our appreciation to the SATNAC review committee for their constructive feedback, and to all colleagues who contributed valuable insights throughout the development and evaluation of this work

REFERENCE

[1] Shah, S.D.A., Gregory, M.A. & Li, S. 2021. Cloud- (Oughton & Jha, 2021) Native Network Slicing Using Software Defined Networking Based Multi-Access Edge Computing: A Survey. *IEEE Access*, 9: 10903–10924.

[2] Kolhar, M., 2021. Zeroize: A new method to improve utilizing 5G networks when running VoIP over IPv6.

[3] Lorincz, J. and Klarin, Z., 2024. A Comprehensive Analysis of the Impact of an Increase in User Devices on the Long-Term Energy Efficiency of 5G Networks. *Smart Cities*, *7*(6), pp.3616-3657.

[4] Ilmudeen, A., 2022. Artificial intelligence, big data analytics and big data processing for IoT-based sensing data. In *Transforming management with AI, big-data, and IoT* (pp. 247-259). Cham: Springer International Publishing.

[5] Merino-Gomez, P., Garcia, B., Andreo, C., Artuñedo, D. and Macias, J., 2024, June. On-demand Trial Networks over 6G-SANDBOX infrastructure. In *2024 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)* (pp. 1021-1026). IEEE.

[6] Jagatheesaperumal, S.K., Rahouti, M., Ahmad, K., Al-Fuqaha, A. and Guizani, M., 2021. The duo of artificial intelligence and big data for industry 4.0: Applications, techniques, challenges, and future research directions. *IEEE Internet of Things Journal*, *9*(15), pp.12861-12885.

[7] Ferrag, M.A., Maglaras, L. and Ahmim, A., 2017. Privacy-preserving schemes for ad hoc social networks: A survey. *IEEE Communications Surveys & Tutorials*, *19*(4), pp.3015-3045.

[8] Esenogho, E., Djouani, K. and Kurien, A.M., 2022. Integrating artificial intelligence Internet of Things and 5G for next-generation smartgrid: A survey of trends challenges and prospect. *Ieee Access*, *10*, pp.4794-4831.

[9] Rafique, W., Barai, J.R., Fapojuwo, A.O. and Krishnamurthy, D., 2024. A survey on beyond 5g network slicing for smart cities applications. *IEEE Communications Surveys & Tutorials*, *27*(1), pp.595-628.

[10] Subedi, P., Alsadoon, A., Prasad, P.W.C., Rehman, S., Giweli, N., Imran, M. and Arif, S., 2021. Network slicing: A next generation 5G perspective. *EURASIP Journal on Wireless Communications and Networking*, *2021*(1), p.102.

[11] Bartsioka, M.L.A., Bartsiokas, I.A., Gkonis, P.K., Papazafeiropoulos, A.K., Kaklamani, D.I. and Venieris, I.S., 2025. A Federated Learning Scheme for Eavesdropper Detection in B5G-IIoT Network Orientations. *IEEE Open Journal of the Communications Society*.

[12] Koppolu, H.K.R., 2024. The Impact of Data Engineering on Service Quality in 5G-Enabled Cable and Media Networks. *European Advanced Journal for Science & Engineering (EAJSE)-p-ISSN 3050-9696 en e-ISSN 3050-970X*, *2*(1).

[13] Adeosun, O.A., Fakeyede, M.O., Adesola, A.A., Akingbulere, G. and Anifowoshe, D.O., 2024. Security Implication of network slicing in 5g-Enabled IoT environment. World Journal of Advanced Research and Reviews, 24(03), pp.2359-2373.

[14] Cunha, J., Ferreira, P., Castro, E.M., Oliveira, P.C., Nicolau, M.J., Núñez, I., Sousa, X.R. and Serôdio, C., 2024. *Enhancing network slicing security:*

[15] Owoko, W., 2024. Exploring the technological advancements and security issues of 5G. *World Journal of Advanced Research and Reviews*, *23*, pp.812-846.

[16] Dangi, R., Jadhav, A., Choudhary, G., Dragoni, N., Mishra, M.K. and Lalwani, P., 2022. Ml-based 5g network slicing security: A comprehensive survey. Future Internet, 14(4), p.116.

[17] Fakhouri, H.N., Alawadi, S., Awaysheh, F.M., Hani, I.B., Alkhalaileh, M. and Hamad, F., 2023. A comprehensive study on the role of machine learning in 5G security: Challenges, technologies, and solutions. Electronics, 12(22), p.4604.

[18] Varadharajan, V., Karmakar, K.K., Tupakula, U. and Hitchens, M., 2021. Toward a trust aware network slice-based service provision in virtualized infrastructures. *IEEE Transactions on Network and Service Management*, *19*(2), pp.1065-1082.

[19] Saeed, M.M., Hasan, M.K., Obaid, A.J., Saeed, R.A., Mokhtar, R.A., Ali, E.S., Akhtaruzzaman, M., Amanlou, S. and Hossain,

A.Z., 2022. A comprehensive review on the users' identity privacy for 5G networks. *IET Communications*, *16*(5), pp.384-399.

[20] Eleftherakis, S., Giustiniano, D. and Kourtellis, N., 2025, June. SoK: Evaluating 5G-Advanced Protocols Against Legacy and Emerging Privacy and Security Attacks. In *18th ACM Conference on Security and Privacy in Wireless and Mobile Networks* (pp. 196-210).

[21] Hamidèche, S.A., 2024. *Machine learning for mobile network user behavior detection in 5G networks and beyond: environment and application classification* (Doctoral dissertation, Université de Rennes).

[22] Kareem, K.M., 2024. The impact of IMSI catcher deployments on cellular network security: challenges and countermeasures in 4G and 5G networks. *arXiv preprint arXiv:2405.00793*.

[23] Dixit, S., Dutta, A., Agrawal, S., Olivas, M.A., Bhatia, V., Carrillo, C.D.A., Jha, P., Jabhera, M., Karna, A., Khanganba, S.P. and Kimambo, C., 2023, November. INGR Roadmap Connecting the Unconnected Chapter. In *2023 IEEE future networks world forum (FNWF)* (pp. 1-88). IEEE.

[24] Ullah, A., 2025. AI Driven Optimization of Resource Allocation and Cost Efficiency in Cloud Computing Environments.

[25] Wang, J., Yan, Z., Wang, H., Li, T. and Pedrycz, W., 2022. A survey on trust models in heterogeneous networks. IEEE Communications Surveys & Tutorials, 24(4), pp.2127-2162.

[26] Zhang, T., Yan, L. and Yang, Y., 2018. Trust evaluation method for clustered wireless sensor networks based on cloud model. Wireless Networks, 24(3), pp.777-797.

[27] Alsadie, D., 2024. Artificial intelligence techniques for securing fog computing environments: trends, challenges, and future directions. IEEE Access.

[28] Hussein, D.H. and Ibnkahla, M., 2025. Towards Intelligent Intent-based Network Slicing for IoT Systems: Enabling Technologies, Challenges, and Vision. IEEE Transactions on Network and Service Management.

[29] Topcu, A., 2023. Intelligent 5G Network Slicing for Vehicular Networks (Doctoral dissertation, University of Leeds).

[30] De Alwis, C., Porambage, P., Dev, K., Gadekallu, T.R. and Liyanage, M., 2023. A survey on network slicing security: Attacks, challenges, solutions and research directions. *IEEE Communications Surveys & Tutorials*, *26*(1), pp.534-570

[31] Alanazi, M.H., 2023. Machine Learning-based Secure 5G Network Slicing: A Systematic Literature Review. International Journal of Advanced Computer Science & Applications, 14(12).

[32] Hu, J. and Wu, J., 2022. 5G network slicing: Methods to support blockchain and reinforcement learning. *Computational Intelligence and Neuroscience*, *2022*(1), p.1164273.

[33] Wong, S., Han, B. and Schotten, H.D., 2022. 5G network slice isolation. *Network*, *2*(1), pp.153-167.

[34] Abood, M.J. and Abdul-Majeed, G.H., 2023. Classification of network slicing threats based on slicing enablers: A survey. International Journal of Intelligent Networks, 4, pp.103-112.

[35] Hamdi, W., Ksouri, C., Bulut, H. and Mosbah, M., 2024. Network slicing-based learning techniques for IoV in 5G and beyond networks. IEEE Communications Surveys & Tutorials, 26(3), pp.1989-2047.

[36] Park, S., Cho, B., Kim, D. and You, I., 2022. Machine learning based signaling ddos detection system for 5g stand alone core network. Applied Sciences, 12(23), p.12456.

[37] Köksal, S., Dalveren, Y., Maiga, B. and Kara, A., 2021. Distributed denial‐of‐service attack mitigation in network functions virtualization‐based 5G networks using management and orchestration. International journal of communication systems, 34(9), p.e4825.

[38] Sheibani, M., Konur, S., Awan, I. and Qureshi, A., 2024. A Multi-layered defence strategy against DDoS attacks in SDN/NFV-based 5G mobile networks. Electronics, 13(8), p.1515.

[39] Sattar, D. and Matrawy, A., 2019, June. Towards secure slicing: Using slice isolation to mitigate DDoS attacks on 5G core network slices. In 2019 IEEE Conference on Communications and Network Security (CNS) (pp. 82-90). IEEE.

[40] Alzhrani, R.M. and Alliheedi, M.A., 2023. 5g networks and iot devices: Mitigating ddos attacks with deep learning techniques. *arXiv preprint arXiv:2311.06938*.

Building a Low-cost Cloud Native 5G Network Slicing Experimental Testbed: Open-Source Solutions, Lessons Learned and Future Directions

Humphrey Owuor Otieno
Electrical Engineering
University of Cape Town
Cape Town, South Africa
otnham001@myuct.ac.za

Lusani Mamushiane
Council for Scientific and Industrial Research (CSIR)
Pretoria, South Africa
lmamushiane@csir.co.za

Chepkoech Maurine
Electrical Engineering
University of Cape Town
Cape Town, South Africa
chpmau001@myuct.ac.za

Tariro Mukute
Electrical Engineering
University of Cape Town
Cape Town, South Africa
mkttar001@myuct.ac.za

Joyce Mwangama
Electrical Engineering
University of Cape Town
Cape Town, South Africa
joyce.mwangama@uct.ac.za

Bessie Malila
Dept. of Electrical and Electronic Engineering Science
University of Johannesburg
Johannesburg, South Africa
bessiem@uj.ac.za

Elena-Ramona Modroiu
Next Generation Networks Dept.
Technische Universität Berlin
Berlin, Germany
elena-ramona.modroiu@tu-berlin.de

Thomas Magedanz
Software-based Networks Dept.
Fraunhofer FOKUS Institute
Berlin, Germany
thomas.magedanz@fokus.fraunhofer.de

***Abstract*—Research on network slicing has gained some ground since its introduction to 5G networking. Network slicing has evolved from ideas, concepts, and simulations to real-world implementations. With this advancement, the realization and experimentation of network slicing remain a challenge due to the complexity involved. Research and experimentation testbeds have been built to explore and advance network slicing. However, most are quite expensive to set up or complicated for the novice researcher exploring 5G network slicing. In this work, we present a comprehensive and repeatable methodology that can be used to realize network slicing. Our approach focuses on building a low-cost, experimental cloud-native testbed using open-source solutions. The testbed leverages containerization and orchestration platforms to deploy the core network. We demonstrate end-to-end network slicing by extending the Open Air Interface's reference setup with custom Helm charts. We validate the process of the user equipment requesting a slice on the network, to admission and management of the slice using slice-specific identifiers like Slice Service Type and Slice Differentiator. We highlight the lessons learned when automating this testbed setup to ensure the testbed is repeatable, consistent and proper resource isolation is achieved per slice as envisioned in network slicing. This work serves as a practical guide for researchers to experiment with and explore network slicing using readily available off-the-shelf hardware.**

***Index Terms*—5G, Network Slicing, Cloud Native, Testbed.**

I. Introduction

The fifth generation of mobile networks (5G) promise a step-change in how connectivity is provisioned and monetised, particularly through network slicing, which enables the creation of multiple, service-level agreement (SLA)-differentiated logical networks over a shared physical infrastructure comprising the radio access network (RAN), transport, and core components [1]. The RAN manages radio spectrum and wireless connectivity to user devices. The core network (CN) handles authentication, session management, and service orchestration, while the transport network (TN) provides high-capacity backhaul, midhaul, and fronthaul connectivity that interconnects these elements. In the African context, where operators must simultaneously extend coverage into sparsely populated or low-average revenue per user (ARPU) regions and unlock enterprise revenue to sustain capital and operational expenditures, slicing offers a pragmatic path to reuse infrastructure across verticals while guaranteeing performance isolation [2].

In 5G, three primary slices are defined, i.e, the ultra-reliable low latency communication (uRLLC), enhanced mobile broadband (eMBB), and the massive machine type communication (mMTC). uRLLC enables mission-critical applications requiring sub-millisecond latency and 99.999% reliability such as industrial automation and autonomous vehicles, eMBB delivers high-throughput broadband services for applications like 4K/8K video streaming and virtual reality, while mMTC supports massive IoT deployments with millions of low-power, low-data-rate sensors and devices per square kilometre, such as IoT-enabled agriculture and smart metering. By so doing, one network build-out is turned into several

monetisable services, over logically separated and isolated network slices, each with distinct QoS requirements [3].

In the realm of experimentation, research and evaluation, E2E network slice implementations are considered complex and add-on services beyond what a basic E2E 5G network offers. Industry-scale testbeds have been built to advance wireless research in Europe and America. They cover city-wide deployment of wireless networks. Often, access to such industry scale testbeds, especially for the African researcher, is still a challenge. Smaller scale 5G mobile network testbeds like [4] [5] discuss low-cost deployment of 5G networks. Such testbeds accelerate capacity building for researchers who want to familiarize with 5G technology, and those with limited access to the industry-scale 5G equipment, or want to go beyond the abstraction provided by the large-scale testbed to fully understand the operation of complex 5G networks while solving practical problems [6]. Additionally, access to such lab-scale testbeds is key in prototyping and research of advanced 5G features that go beyond standard deployment and can later be extended to large wireless networks and very niche use cases. E2E 5G network slicing is one of these advanced features of the deployment of the 5G network. This is because of the complexity that comes with the logical separation and management of resources and access of the network across the different domains from the RAN, TN and CN. Esmaeily et al. [7] present a design criteria for setting up small scale 5G testbeds for network slicing. The key attributes defined and identified to set up a small testbed for network slicing are: Support for enabling technologies like software defined networking (SDN), network function virtualisation (NFV) and cloud computing that allows for flexible, dynamic and programmable orchestration of the network at different levels. The 5G testbed should offer multi-domain support that spans across the RAN, TN, and CN which in turn provides the possibility of offering E2E network slicing.

Open-source ecosystems like the OpenAirInterface (OAI) project, which comprises the OAI Core Network [1] and OAI RAN [2], along with software radio system (srs) RAN [3], and Open5GS [4] offer a disaggregated discussion that explains how network slicing can be achieved. Despite these software modules being among the popular open-source frameworks deployed in experimental research of 5G testbeds, they still fall short of offering a clear, detailed approach for implementing E2E 5G network slicing. OAI, in their documentation [5], give a scenario of 5G network slicing at the CN and possibilities of implementing slicing on the RAN, but a detailed explanation of how this can be configured and achieved in their newly built cloud native CN, that also takes in YAML configurations and is deployed using HELM package manager and K8 resource orchestrator, remains limited and not very descriptive.

Therefore, the aim of this paper is to provide a practical approach of setting up a low cost experimental testbed for evaluating network slicing in 5G. We summarize the contribution of this work as follows:

- We describe the process of setting up a cloud native small-scale testbed for rapid research, experimentation, and capacity building using low-cost devices. In addition, we demonstrate the separation of the control plane and data planes in the testbed. The control plane handles all signaling, management, and session coordination functions, while the data plane is responsible for the actual data transmission including packet formarding and quality of service control. In a 5G testbed environments, the separation of control and data planes allows for more granular testing, better resource utilization, and the ability to experiment with different deployment models that reflect real-world 5G network scenarios.
- We present the implementation steps that achieve 5G network slicing to showcases two user equipment assigned to dedicated slice. At its base, this work builds upon OAI configuration and design guides and extends the deployment of network slices by translating the docker compose methodology in [6] into helm charts to suit our custom slicing needs. Our approach aims to maintain simplicity while demonstrating and documenting the technical aspects of what is needed to implement and have an advanced automated cloud native 5G testbed.

The rest of the paper is organized as follows; in section II, we highlight a background on the 5G reference architecture in relation to network slicing and the cloud native technology stack that can be used to realize network slicing. In section III, we present some of the research and works done in the topic of 5G testbeds and network slicing, which serve as a foundation on which this work is built. The description of steps involved to setup the testbed are discussed in section IV. In section V, we validate the realization of network slicing on the testbed, and in section VI we highlight key lessons learned in the implementation, and thereafter we provide future research directions in section VII. Finally, we conclude this work in section VIII.

II. Background

A. General 5G System architecture in the context of Network Slicing.

The 5G system includes the 5G CN and 5G RAN that are interconnected by the N2 and N3 interfaces. The N2 interface is the entry point of the 5G RAN into the 5G Core via the access and management function (AMF). The N3 interface is the data plane that carries user traffic from the RAN through the CN and out to external Data Networks (DNs) via the N6 interface. Within the 5G CN, the different network functions expose a service based interface, which can be used

[1] https://gitlab.eurecom.fr/oai/cn5g

[2] https://gitlab.eurecom.fr/oai/openairinterface5g

[3] https://github.com/srsran/srsRAN_Project

[4] https://github.com/open5gs/open5gs

[5] https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed/-/blob/master/docs/DEPLOY_SA5G_SLICING.md

[6] https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed/-/blob/master/docs/DEPLOY_SA5G_SLICING.md

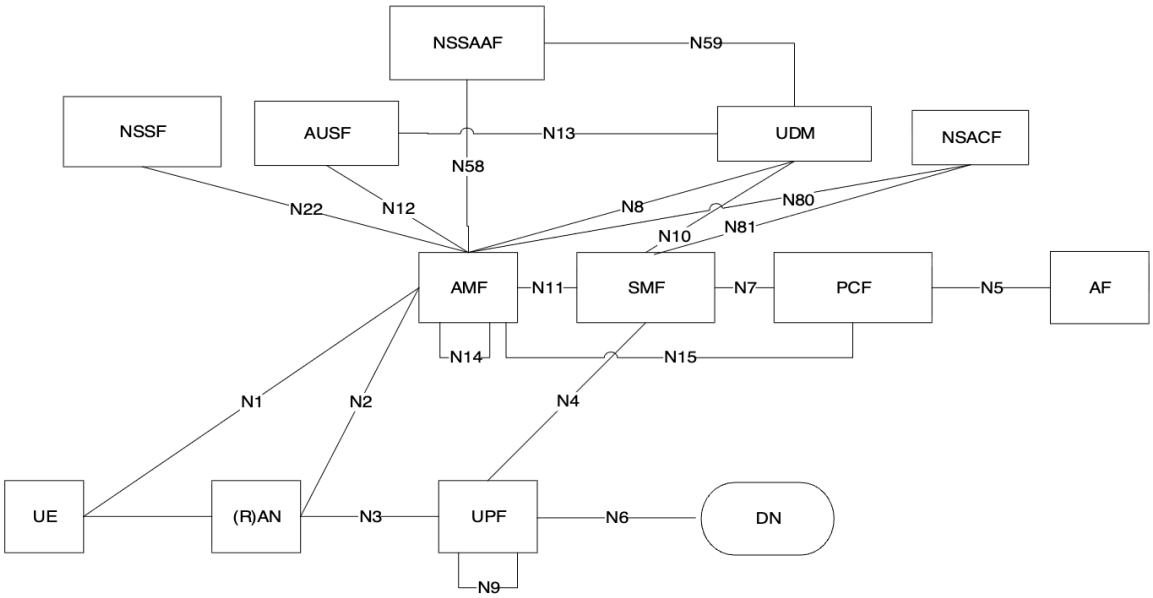


Fig. 1: 5G System Architecture and Interfaces [8]

by different CN functions to access the services provided by other NFs while also allowing multiple interactions among the NFs defined by a specific network slice [9].

It is important to note that the service-based architecture that the 5G CN employs for the different NFs is complemented by the implementation of NFs as VNFs, which allows software NFs to be implemented either as virtual machines and containers on Docker or Kubernetes.

The NRF provides and maintains the information of all the deployed NF instances to support service discovery. In the context of network slicing, the NSSF manages network slice selection and access to network slices in coordination with the AMF and the SMF via the N22 interface [9]. The Slice/Service Type (SST) and Session Description (SD) identifiers are crucial in a 5G network to ensure that slices are described, configured and isolated [10], [11]. 3GPP defines SST values as 1 for eMBB, 2 for URLLC, and 3 for mMTC, with 4–127 reserved and 128–255 operator-specific [12], while the SD is a 24-bit value that distinguishes slices of the same SST, ensuring the SST+SD pair uniquely identifies a network slice instance. They also act as identifiers that can be paired with a DNN value to ensure strict isolation on network slices at the CN. Furthermore, as noted by Limani et al [11], slice isolation can be achieved by allocating both physical and logical resources for a slice. For instance, the SMF, which is responsible for UE IP allocation, UPF selection and session management, can be configured to only issue an IP address pool to a specific DNN per slice to ensure logical isolation of traffic at the CN over distinct networks.

The 5G CN can also be separated into the control plane and data plane. The control plane carries control plane information and signalling messages while the data plane carries user traffic [10]. A comprehensive 5G system architecture with the associated interfaces and their interaction is shown in Figure 1

B. Cloud Native Technology Stack Supporting 5G Network Slicing

In 5G network slicing, orchestrating and managing logical networks on top of a shared physical infrastructure requires flexibility and efficiency [13]. This flexibility in network slicing use cases means service elasticity, service upgrade and update to support the life cycle of a network slice. This life cycle involves slice creation, execution, monitoring and termination [13]. This flexibility and elasticity in the context of 5G networking can be achieved by softwarization, virtualisation, and cloud computing technologies supporting the deployment of 5G network functions.

A few technologies have been used to aid the deployment and management of 5G network functions. Infrastructure management, from bare metal to initial configuration and setup, can be managed and automated through Infrastructure as Code principles, for example using Ansible [7]. Ansible playbooks ensure that the infrastructure used to deploy 5G networks is idempotent, meaning running the same playbook multiple times will give the same results. In the case of 5G testbeds, this ensures that a consistent desired state of the underlying testbed is maintained. A private 5G experimentation testbed built using infrastructure as code with Ansible is demonstrated in [14].

Beyond initial testbed set up and base configuration, 5G RAN and CN NFs need to be deployed and tied together

[7]https://docs.ansible.com/ansible/latest/getting_started/introduction.html

to create an end-to-end 5G network. With softwarization of NFs, a cloud native approach with relevant tools can be used to implement network slicing. The design patterns of cloud native applications discussed in [15] involve containerization, where an application, in this case 5G NFs, are packaged as lightweight bundles of applications code, libraries, configuration and dependencies that ensures applications run consistently over different computing environments. Containerization can be achieved using tools such as Docker [8] to build a Docker image using a Dockerfile [16]. In the realm of 5G NFs, there are a number of open source 5G network functions, from the RAN to the CN, that have been built and stored on different container registries such as Docker Hub [9] including implementations from OAI [10] and Open5GS [11].

The next design pattern of cloud native applications that can support network slicing is the composition of microservices to create and serve an application [15]. In the context of network slicing, multiple container network functions must be interconnected to provide an end-to-end 5G network. As the number of containers grows, efficient management and orchestration of such containers becomes a challenging. Kubernetes [12], an open-source container deployment and management systems, addresses this by offering extra services such as lifecycle management, updating and health check monitoring and application recovery to realize flexibility and efficiency in 5G network slicing. Furthermore, Kubernetes also supports resource isolation capabilities through namespaces, which is key in network slicing isolation. In later sections, we leverage these capabilities to deploy network slices in the CN of the testbed proposed in this work.

In the context of 5G network slicing, infrastructure management can be done through Ansible, while containerized 5G NFs are orchestrated on Docker Engine. Automation is needed to manage complex Kubernetes application configuration and distribution. Helm [13], the Kubernetes package manager, makes it possible to manage a Kubernetes application configuration file, install a new application that can be defined using a Helm chart, and update a Kubernetes application through the Helm command line tool. Therefore, in 5G network slicing, the life cycle of a network slice from slice creation, execution, monitoring and termination can be managed by the technology stack made up of Ansible, Docker, Kubernetes and Helm. We demonstrate this implementation in our work of building a simplified, low-cost network slicing-focused testbed in section IV.

III. Related Works

Large-scale city-wide testbeds with a focus on either next-generation wireless communication or network slicing have been implemented in [17] and [18]. Large-scale testbeds can be used for rigorous real-world deployments and evaluations. At the same time, there is a desire to test the specific low-level features and capabilities of novel designs and architectures of 5G networks to achieve the vision of the 3GPP 5G system specifications [19], [20]. As such, prototyping systems that implement or are biased towards individual parts of the 5G of the mobile network architecture, such as the RAN, TN or CN, miss out on the opportunity to evaluate complex end-to-end multitenancy networks that evaluate implementations, such as network slicing [5].

The work by [7] implements 5GIIK, which is a small-scale testbed for E2E network slicing with the capability of management and orchestration of network resources. As much as they also use the cloud native philosophy to deploy their network slices, we view this work as an early contribution to the initial approaches of building a network slicing testbed. While this contribution is valuable, it is built on 4G LTE RAT using srsLTE[14] and OAI CN and therefore does not align with 3GPP standards for defining and instantiating a network slice. The authors follow a hierarchical approach for creating slices, beginning with VNF Descriptors (VNFDs) as the first level, followed by Network Service Descriptors (NSDs) as the second level, and finally the Network Slice Instance Descriptor (NSID) which integrate the service instances to form a complete network slice.

Despite this VNF-focused description approach of creating an end-to-end network slice on this testbed, the 3GPP standard procedure of creating and adhering to 5G network slicing from identification and life cycle is not followed. Furthermore, we do not see 5G UE attach, and resource requests on the network for different slices that are available. This is an aspect that we demonstrate in our work and we employ standard identifiers, SST and SD values, to define, label, create, and monitor network slice instances.

Recent work by [21] demonstrates the lack of true end-to-end network slicing. The authors implement network slices using open-source software. They use srsRAN for LTE implementation at the RAN and Open5GS as the core network. Their infrastructure is managed using OpenStack, Opendaylight (ODL), and Open-Source MANO (OSM). A B210 USRP SDR is used to act as a gNB and propagate wireless signals. Their implementation of network slices involves altering the placement of the UPF to achieve low-latency communication. Although they succeed in this, their approach does not show placement of traffic across the full path of an E2E network slice from the RAN to the CN, nor does it incorporate standardized slice identifiers such as SST and SD values. We can infer this because the RAN was implemented using srsRAN LTE software, which does not support 5G-NR and is unable to use SST and SD values for the authorization of access to the network.

The documentation by OAI [15] on network slicing is the closest work related to this paper. We base our implementation

[8] https://docs.docker.com/get-started/docker-overview/
[9] https://hub.docker.com/
[10] https://hub.docker.com/u/oaisoftwarealliance
[11] https://hub.docker.com/r/openverso/open5gs/tags
[12] https://kubernetes.io/docs/concepts/overview/
[13] https://helm.sh/

[14] https://github.com/srslte
[15] https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed/-/blob/master/docs/DEPLOY_SA5G_SLICING.md

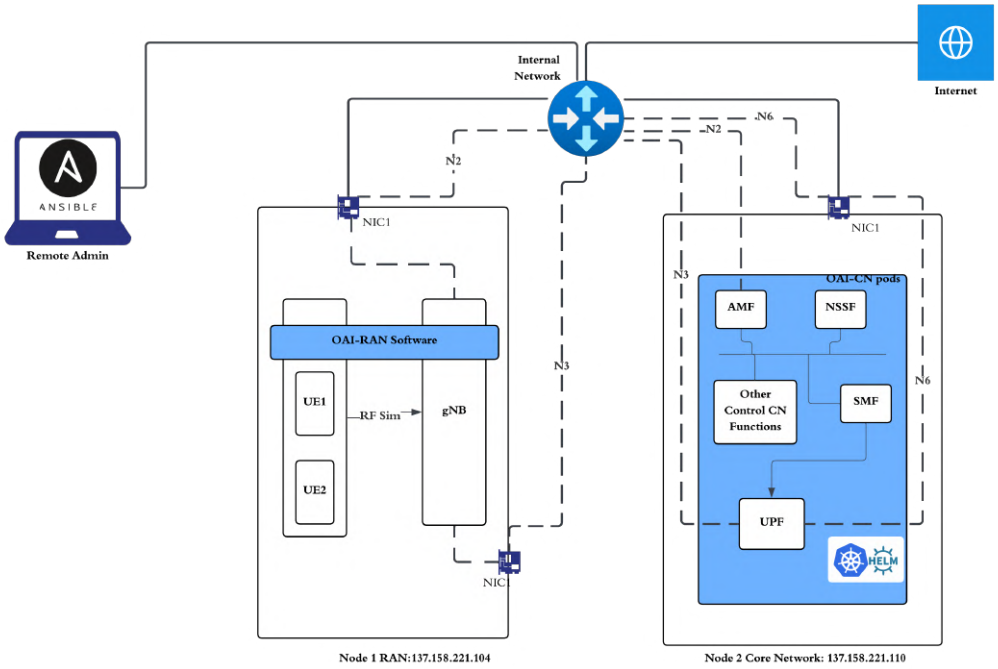


Fig. 2: Proposed 5G network slicing testbed architecture.

on the basic Helm charts provided by the OAI project, which we extend and refined to ensure that there is total separation of the control plane and data plane in our network slice implementation. In addition, it is important to note that their network slice implementation involved two slices sharing an NRF, and one dedicated NRF for the third slice. The only common control functions shared by the three slices were the NSSF, AMF, UDM,UDR, and AUSF. Orchestration and management of the slices is based on Docker compose, which in our work we implement total separation of control plane and data plane, where a dedicated data plane serves each slice with a shared control plane. This is done by translating the Docker Compose startup scripts into custom Helm charts on Kubernetes to manage the network slice infrastructure in our testbed.

This paper also presents an easy-to-follow step-by-step configuration to replicate this work in low-cost off-the-shelf computing devices for experimentation and research.

IV. Testbed Setup

The testbed consists of two desktops running Ubuntu 22.04. The machine hosting the core network features an Intel Core i7-9700F processor (8 cores), 24 GB of DDR4 RAM, an NVIDIA GeForce GT 710 GPU, and a 1 TB Seagate HDD. The RAN computer features an Intel Core i7-9700F processor (8 cores), 32 GB of DDR4 RAM, an NVIDIA GeForce GT 710 GPU, and a 1 TB Seagate HDD. The RAN machine runs the OAI RAN software suite that implements both the gNB and the nr-ue. We connect the two using an RF simulator. We did not use real UEs in this setup because, our primary objective was to evaluate an E2E network slicing deployment which extends beyond the basic connectivity testing. E2E connectivity tests using real UEs and different software suites from srsRAN, OAI and Open5GS have been evaluated in [4], [22], [23] and [24]. In these testbeds, the focus was not on evaluating network slicing but basic E2E connectivity, experience profiling and RAN optimization in [23] and [24], respectively, and the building of a framework for granular data collection and data set building in [22]. As we advance towards network slicing in this work, the limitation that made it difficult to use real UEs is UE capabilities for network slice selection using SST and SD values. Therefore, we evaluate the slicing-focused testbed using emulated UEs.

A. Initial Infrastructure Setup

We took the approach of setting up the infrastructure of the current testbed from bare metal to a basic setup with all software modules set and configured using the infrastructure as code paradigm with Ansible. The Ansible playbooks that we developed aimed to achieve automated RAN node setup for both emulated UE tests and real UE tests. To this end, we implemented playbooks that fetch, install and set up the USRP Hardware Driver (UHD) for SDR connections. On the RAN node, we implement a playbook that installs OAI RAN software suite, executes the CPU performance optimization scripts and launches the OAI gNB. For the CN node, we developed Ansible playbooks to allow the deployment of the OAI-CN software in a cloud native manner. The playbooks

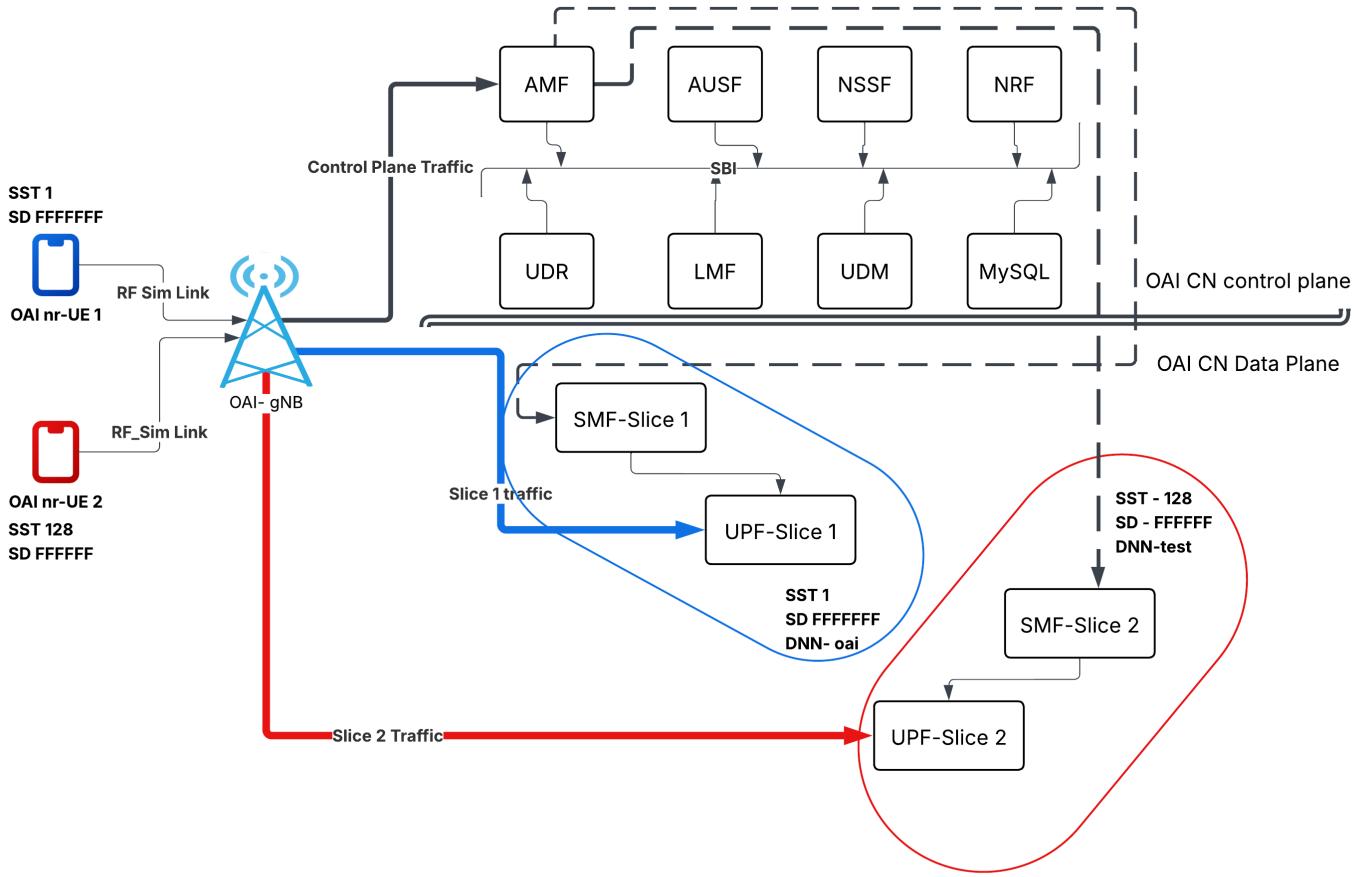


Fig. 3: Network slicing testbed implementation and architecture

fetch and install Docker, Kubernetes with MicroK8s and Multus Meta CNI and HELM for package management. This Infrastrcure-as-Code approach ensures that testbed setup is repeatable, consistent and Idempotent at all times. The Ansible playbook files used for this setup are publicly available in this Github repository [16]

B. Network Configuration

This testbed was configured using OAI 5GNR and OAI-CN. The OAI RAN implementation (both the gNB and the UE) was based on an RF simulator to interconnect the two components. This is because of the limitation of real UEs that would allow the configuration of slice identifiers i.e., the SST and SD values. For the evaluation of our setup, we configured two slice identifiers: Slice 1 with SST 1, SD FFFFFF, and DNN oai, Slice 2 with SST 128, SD FFFFFF, and DNN test. These slice identifiers were cascaded from the UE configuration files through to the CN configuration files.While we relied on the Helm chart configuration files provided by OAI as a baseline, the modification made to customize this deployment in our testbed was to create three different Helm charts and accompanying configuration files that will be picked up whenever the charts were installed to implement and run this testbed. With all these Helm charts, we created different namespaces for the control plane network functions, and for the data plane serving each slice instance.

Connecting the RAN to CN is important for realizing both E2E network connectivity as well as E2E networks slice implementation. Since we have deployed the CN as Kubernetes pods, we needed to expose the AMF in the control plane and the UPF which constitutes the data plane in the CN. The other CN components are connected using the SBI interface implemented by the private addresses within the Kubernetes network. To achieve this, we used Multus, a Container Network Interface (CNI) plugin which enabled the AMF and UPFs to bind to our local network addresses for reachability by the gNB. Accordingly, the N2 interface, N3 interfaces on UPF-slice 1 and UPF-Slice 2 were configured to bind to the Multus plugin in the ”values.yaml”

[16]https://github.com/HumpreyDev/5G-CN-Slicing

configuration files that each helm chart relied on during deployment. Furthmore, since the control plane and data planes we configured in this testbed had different namespaces, we were required to make adjustments in the configuration file to ensure that the NRF would easily register the two UPFs and SMFs that are in different namespaces and allow discovery of these network functions by the NSSF and the AMF during registration and network slice selection procedure. We achieved this by configuring the Go templates of the SMF and UPF. Talk about NSSF configuration to show allowed slices

V. Results and Validation

In this section, we demonstrate the implementation of network slices on the testbed. This demonstration shows the process from slice request, slice selection and network slice implementation by the different associated network functions. As mentioned earlier, we have configured two UEs to connect to the network slices with the following IDs, SST 1 SD FFFFFF and DNN oai for Slice 1 and SST 128, SD FFFFFF and DNN test for Slice 2. The configurations for UE1 and UE2 are shown in Fig 4 and Fig 5.

```
uicc0 = {
imsi = "001010000000100";
key = "fec86ba6eb707ed08905757b1bb44b8f";
opc= "C42449363BBAD02B66D16BC975D77CC1";
dnn= "oai";
nssai_sst=1;
nssai_sd=0xffffff;
}
```

Fig. 4: UE 1 slice and DNN configuration.

```
uicc0 = {
imsi = "001010000000101";
key = "fec86ba6eb707ed08905757b1bb44b8f";
opc= "C42449363BBAD02B66D16BC975D77CC1";
dnn= "test";
nssai_sst=128;
nssai_sd=0xffffff;
}
```

Fig. 5: UE 2 slice and DNN configuration.

A. RAN Network Slice Request to the CN

The initial request is sent to the AMF from the gNB by the UE to request a slice. In this case, the AMF will receive two slice requests, SST 1 SD FFFFFF for Slice 1 and SST 128, SD FFFFFF for Slice 2. These requests are then forwarded to the NSSF to ensure that these slices are configured as shown in the Fig. 6.

If these slices are not configured in the NSSF configuration file, the slices will not be allowed and the AMF will not process the UE request further. Moreover, an error will be returned to the UE. Fig. 7 and Fig. 8 show the NSSF–AMF

```
# Reference:- TS 29.531 R16.0.0., Section- 6.1.6 Data Model
info:
  version: 2.0.0
  description: OAI-NSSF Release v2.1.0
configuration:
  nsiInfoList:
    - snssai:
        sst: 1
        sd: 'FFFFFF'
      nsiInformationList:
        nrfId: http://oai-nrf:80/nnrf-disc/v1/nf-instances
        nsiId: '100'
        nrfNfMgtUri: http://oai-nrf:80/nnrf-nfm/v1/nf-instances
    - snssai:
        sst: 128
        sd: 'FFFFFF'
      nsiInformationList:
        nrfId: http://oai-nrf:80/nnrf-disc/v1/nf-instances
        nsiId: '10'
  taInfoList:
    - tai:
        plmnId:
          mcc: '001'
          mnc: '01'
        tac: '0x0001'
      supportedSnssaiList:
      - sst: 1
        sd: 'FFFFFF'
      - sst: 128
        sd: 'FFFFFF'
```

Fig. 6: Redacted NSSF configuration file

interaction log messages for slices that are correctly configured and allowed by the NSSF. The NSSF then responds to this NSSAI request from the AMF by checking if the requested slice by the UE is among the allowed slice in the CN. If these values match, the NSSF responds with a valid S-NSSAI for the NSI and Authorized network slice as shown in Fig. 7 and Fig. 8.

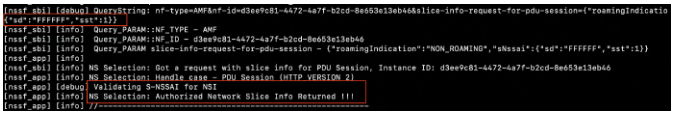


Fig. 7: AMF and NSSF Authorized Slice 1 Request and Response

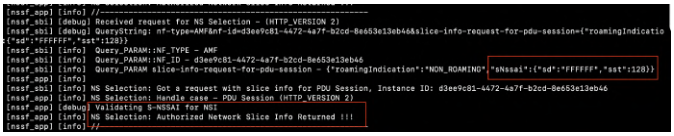


Fig. 8: AMF and NSSF Authorized Slice 2 Request and Response

B. CN functions slice selection

It is important to note that in the CN configuration files, we explicitly configured SMF selection at the AMF. This means, the AMF, in conjuction with the NRF will select the appropriate SMF based on the S-NSSAI. In our case, since

we have two slices configured in the network, we expect the AMF to select the appropriate SMF and send the PDU session setup request.

The SMF also engages in a UPF selection criteria based on the S-NSSAI information and slice request it gets from the AMF. And build a UPF graph. This selection based on SST 1 SD FFFFFF and SST 128, SD FFFFFF is shown in the Fig 9 and Fig 10.



Fig. 9: SMF selection by AMF for Slice 1

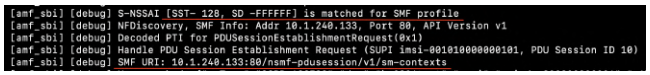


Fig. 10: SMF selection by AMF for Slice 2

To further verify that different SMFs were selected by the AMF for the two slices configured in the network, in Fig. 9 and Fig. 10, the SMF URI is different for each selection procedure. Slice 1 SMF IP address is 10.1.240.160 while Slice 2 SMF IP address is 10.1.240.133.

C. Network Slice Establishment

After SMF selection, the SMF sends the PDU session creation request to the appropriate UPF serving either the SST 1 , SD FFFFFF slice instance or the SST 128, SD FFFFFF slice instance. Successful session establishment is confirmed by assigning an IP address to the UE from the subnet associated with the selected network slice instance. We demonstrate the UPF selection procedure for the different NSI in Fig 11 and Fig 12.

It is equally important to note that the UPFs selected by the SMF for each slice instance are assigned different IP addresses. This demostrates UPF selection based on NSSAI (SST and SD pair) information provided by the UE to create a PDU session. The UPF serving slice 1 is on IP address 10.1.240.174 while the UPF serving slice 2 is on IP address 10.1.240.153.

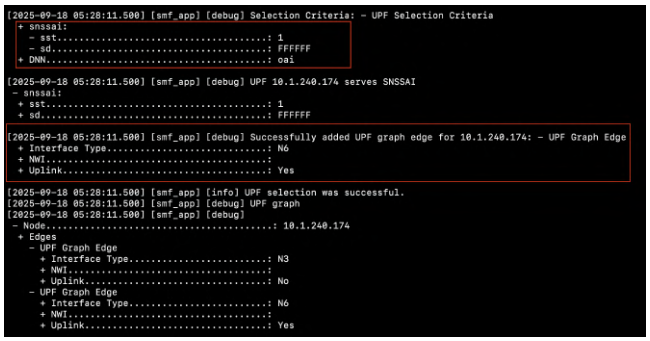


Fig. 11: UPF selection Slice 1

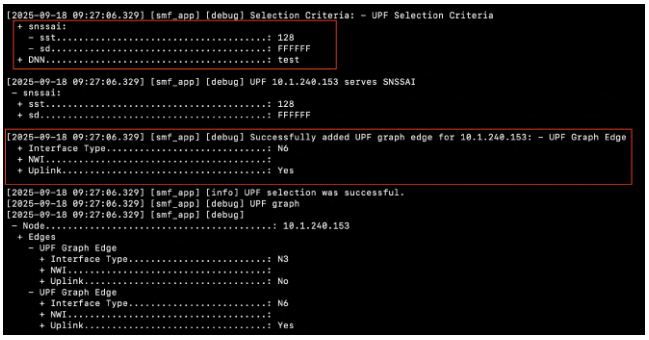


Fig. 12: UPF selection Slice 1

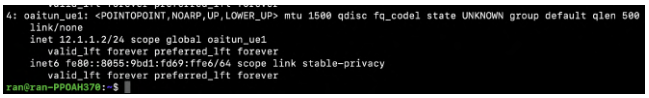


Fig. 13: UE 1 PDU session IP address allocation

In 5G networks, after the succesful authentication which includes network slice NSSAI verification, the SMF triggers session establishment by assigning IP addresses to the UE. In this work, to further ilustrate resource and traffic isolation in the network slices configured, we configured the different SMFs to assign IP addresses to the different UEs with diffrenet SST, SD and DNN values. The subnet configured SST 1, SD FFFFFF and DNN oai was 12.1.1.0/24, while SST 128, SD FFFFFF and DNN test was 14.1.1.0/24. These configurations were passed as config files in the different Helm charts that deployed the SMF and UPF data planes for slice 1 and slice 2. Thereafter, the associated GTPU tunnels are created between the gNB and the UPF through N3. The UEs will use their respective IP address to communicate and send traffic to different subnets in the network. Figures 13 and 14 show the different subnets the two UEs belong to.



Fig. 14: UE 2 PDU session IP address allocation

VI. Lessons Learned

From the implementation of this testbed, several key lessons were learned. First, while open-source software such as OAI and Open5GS provides the foundation for 5G slicing experimentation, their documentation and reference implementations often lack sufficient detail for end-to-end slice realisation, requiring significant customization. Second, Infrastructure-as-Code approaches such as Ansible and container orchestration with Kubernetes and Helm greatly improved repeatability, consistency, and modularity, but required careful tuning of configuration files, namespaces, and templates to ensure proper NF registration and discovery. Third, achieving strict separation of control and data planes

proved essential for validating slice isolation, though it introduced additional complexity in network configuration and management. Finally, the absence of real UEs with slice-aware capabilities (SST and SD configurations) remains a limitation, underscoring the importance of emulation in early-stage prototyping while highlighting the gap toward fully standard-compliant E2E slicing evaluation.

VII. Future Direction

Future work will extend this testbed in several directions. First, we plan to incorporate real 5G UEs with slice-aware capabilities to validate slice selection and service differentiation beyond emulated environments. Second, integration of the transport network/backhaul will be explored to achieve full RAN–TN–CN E2E slicing. Third, we plan to integrate advanced orchestration frameworks such as Open Source Mano [17] will be evaluated on top of our Kubernetes–Helm stack to enable dynamic slice life cycle management, SLA enforcement, and automated scaling. Furthermore, benchmarking the performance of uRLLC, eMBB, and mMTC slice types under varying workloads will provide quantitative insights into resource isolation and QoS guarantees. Finally, we aim to eventually extend this testbed to multi-site or federated deployments across Africa, enabling collaborative and Afrocentric/localised research and experimentation on 5G and beyond networks in low-cost, resource-constrained settings. Specifically, we plan to extend this work to a contribution in the DigitAfrica project[18] that aims to design a Pan Africa led research infrastructure in digital sciences. This work on network slicing targets one of the blueprints that will be developed in DigitAfrica for 5G and 6G experimentation modeling the Slice-RI Post-5G blueprint [19].

VIII. Conclusion

This paper presented a low-cost, cloud-native 5G testbed for end-to-end network slicing. By extending OAI's reference setup with custom Helm charts, we achieved control–data plane separation, slice instantiation, and traffic isolation using SST and SD identifiers. The testbed demonstrates that advanced slicing can be realized on off-the-shelf hardware, offering a practical platform for experimentation and capacity building. Future work will integrate real slice-aware UEs and extend slicing to the transport domain.

References

[1] L. U. Khan, I. Yaqoob, N. H. Tran, Z. Han, and C. S. Hong, "Network slicing: Recent advances, taxonomy, requirements, and open research challenges," *IEEE Access*, vol. 8, pp. 36 009–36 028, 2020.

[2] L. Banda, M. Mzyece, and F. Mekuria, "5g business models for mobile network operators—a survey," *Ieee Access*, vol. 10, pp. 94 851–94 886, 2022.

[3] S. Zhang, "An overview of network slicing for 5g," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 111–117, 2019.

[4] M. Chepkoech, N. Mombeshora, B. Malila, and J. Mwangama, "Evaluation of open-source mobile network software stacks: A guide to low-cost deployment of 5g testbeds," *2023 18th Wireless On-Demand Network Systems and Services Conference, WONS 2023*, pp. 56–63, 2023.

[5] X. Foukas, M. K. Marina, F. Sardis, M. A. Lema, F. Foster, and M. Dohler, "Experience building a prototype 5g testbed," in *EM-5G 2018 - Proceedings of the 2018 Workshop on Experimentation and Measurements in 5G, Part of CoNEXT 2018*. Association for Computing Machinery, Inc, 12 2018, pp. 13–18.

[6] L. A. Phan, D. Pesch, U. Roedig, and C. J. Sreenan, "Building a 5g core network testbed: Open-source solutions, lessons learned, and research directions," in *International Conference on Information Networking*. IEEE Computer Society, 2024, pp. 641–646.

[7] A. Esmaeily, K. Kralevska, and D. Gligoroski, "A cloud-based sdn/nfv testbed for end-to-end network slicing in 4g/5g," in *2020 6th IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2020, pp. 29–35.

[8] TSGS, "Ts 123 501 - v18.5.0 - 5g; system architecture for the 5g system (5gs) (3gpp ts 23.501 version 18.5.0 release 18)," Tech. Rep., 2024. [Online]. Available: https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

[9] R. Trivisonno, X. An, and Q. Wei, "Network slicing for 5g systems: A review from an architecture and standardization perspective," in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2017, pp. 36–41.

[10] A. Shorov, "5g testbed development for network slicing evaluation," in *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. IEEE, 2019, pp. 39–44.

[11] X. Limani, A. Troch, C. C. Chen, C. Y. Chang, A. Gavrielides, M. Camelo, J. M. Marquez-Barja, and N. Slamnik-Kriještorac, "Optimizing 5g network slicing: An end-to-end approach with isolation principles," in *2024 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2024*. Institute of Electrical and Electronics Engineers Inc., 2024.

[12] 3rd Generation Partnership Project (3GPP), "Numbering, addressing and identification," 3GPP, Technical Specification TS 23.003, 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123000_123099/123003/18.06.00_60/ts_123003v180600p.pdf

[13] F. H. Grings, L. B. D. Silveira, K. V. Cardoso, S. Correa, L. R. Prade, and C. B. Both, "Full dynamic orchestration in 5g core network slicing over a cloud-native platform," in *Proceedings - IEEE Global Communications Conference, GLOBECOM*, 2022, pp. 2885–2890.

[14] R. D. B. Lira, L. V. Monteiro, V. S. Simão, L. C. D. Almeida, R. D. Gomes, O. D. A. R. Neto, and P. D. Maciel, "Enabling private 5g experimentation with network programmability and infrastructure as code." Institute of Electrical and Electronics Engineers (IEEE), 4 2025, pp. 1–2.

[15] O. Arouk and N. Nikaein, "Kube5g: A cloud-native 5g service platform," in *Proceedings - IEEE Global Communications Conference, GLOBECOM*, vol. 2020-January, 12 2020.

[16] ——, "5g cloud-native: Network management & automation," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020, pp. 1–2.

[17] S. Fdida, N. Makris, T. Korakis, R. Bruno, A. Passarella, P. Andreou, B. Belter, C. Crettaz, W. Dabbous, Y. Demchenko *et al.*, "Slices, a scientific instrument for the networking community," *Computer Communications*, vol. 193, pp. 189–203, 2022.

[18] D. Raychaudhuri, I. Seskar, G. Zussman, T. Korakis, D. Kilper, T. Chen, J. Kolodziejski, M. Sherman, Z. Kostic, X. Gu *et al.*, "Challenge: Cosmos: A city-scale programmable testbed for experimentation with advanced wireless," in *Proceedings of the 26th annual international conference on mobile computing and networking*, 2020, pp. 1–13.

[19] 3rd Generation Partnership Project (3GPP), "System Architecture for the 5G System (5GS); Stage 2," 3GPP, Technical Specification TS 23.501, 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/18.06.00_60/ts_123501v180600p.pdf

[20] ——, "Procedures for the 5G System (5GS); Stage 2," 3GPP, Technical Specification TS 23.502, 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123502/18.05.00_60/ts_123502v180500p.pdf

[21] P. Vanichchanunt, O. Ritruechai, N. Wuttiananchai, P. Thossaporn, L. Wuttisittikulkij, and S. Paripurana, "Implementation of 5g network slicing using open source software," in *Proceeding - 12th International*

[17] https://www.etsi.org/technologies/open-source-mano

[18] https://www.digitafrica.eu/

[19] https://doc.slices-ri.eu/BlueprintServices/beyond5G/beyond5G.html

Electrical Engineering Congress: Smart Factory and Intelligent Technology for Tomorrow, iEECON 2024. Institute of Electrical and Electronics Engineers Inc., 2024.

[22] H. O. Otieno, J. Mwangama, and B. Malila, "A framework for building application-level end-to-end 5g networking datasets." 2024.

[23] L. Mamushiane, A. A. Lysko, J. Mwangama, H. I. Kobo, and H. O. Otieno, "Experience in profiling and optimizing a 5g standalone radio access network (ran) based on an open source testbed," 2023.

[24] L. Mamushiane, A. Lysko, H. Kobo, and J. Mwangama, "Deploying a stable 5g sa testbed using srsran and open5gs: Ue integration and troubleshooting towards network slicing," in *6th International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems, icABCD 2023 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., 2023.

Data Centers, Cloud and Edge Computing

Digital Twin Implementation for Real-time Structural Health Monitoring

Filmon Gebremichael
Department of Electrical and Electronic Engineering Sciences
University of Johannesburg
Johannesburg, South Africa
201323471@student.uj.ac.za

Michael Grobler
Department of Electrical and Electronic Engineering Sciences
University of Johannesburg
Johannesburg, South Africa
michaelg@uj.ac.za

Sune Von Solms
Department of Electrical and Electronic Engineering Sciences
University of Johannesburg
Johannesburg, South Africa
svonsolms@uj.ac.za

Abstract— **Smart cities require infrastructure monitoring to ensure public safety, particularly with 80% of Johannesburg's bridges in poor condition, making accessible structural health monitoring critical. This paper presents the implementation of a comprehensive digital twin system for structural health monitoring (SHM) utilising fibre Bragg grating (FBG) sensors integrated with modern digital technology stacks. Building upon the evolution of strain sensing technologies over the past century—from Lord Kelvin's 1856 discovery of piezoresistive effects to today's sophisticated fibre optic systems—this paper demonstrates a complete end-to-end solution that addresses the critical gap between traditional qualitative infrastructure assessment and continuous quantitative monitoring. The system deployed on the B2LAB-B3LAB pedestrian bridge at the University of Johannesburg comprises 10 FBG sensors across two channels, a high-performance data acquisition system, and a cloud-based monitoring platform processing 1.72 million data points daily. The implementation features real-time data streaming via WebSocket connections, edge computing through Cloudflare infrastructure, and a comprehensive web dashboard built with Next.js and React. With multiple programming languages, the system achieves real-time monitoring whilst maintaining compliance with some important IEEE 2067-2021, IEEE 1451.0-2024, and ISO/IEC/IEEE 42020:2019 standards. This research contributes to UN Sustainable Development Goal 9 by demonstrating how modern digital infrastructures can enhance the resilience and sustainability of critical infrastructure. The main contribution of this paper is democratising professional-grade FBG sensing through open-source web technologies, enabling resource-constrained municipalities to implement smart infrastructure monitoring.**

Keywords— **structural health monitoring, advanced sensing, digital twin, fibre Bragg gratings, real-time monitoring, cloud computing, WebSocket, edge computing, data analytics, Industry 4.0, smart cities**

I. Introduction

The progressive advancement of sensing technologies and digital infrastructure has fundamentally transformed the landscape of structural health monitoring (SHM), enabling a paradigm shift from reactive maintenance to proactive, data-driven infrastructure management [1]. This transformation builds upon a century of strain-sensing innovation, beginning with Lord Kelvin's demonstration to the Royal Society of London in 1856 that electrical resistance in metallic conductors changes when exposed to mechanical strain [2]. However, it took 82 years before Edward E. Simmons and Arthur C. Ruge independently invented the practical resistive strain gauge in 1938, marking the birth of modern strain-based SHM [2].

This transformation is particularly crucial for smart cities, where infrastructure systems must adapt to increasing urbanisation, climate change impacts, and citizen safety requirements [3], [4]. Smart city initiatives globally recognise advanced sensing networks as fundamental enablers of urban resilience, with structural health monitoring identified as a critical component of intelligent infrastructure management [5]. Yet the high cost and complexity of traditional monitoring systems create a digital divide between well-resourced and developing cities. As Sony et al. [3] note, next-generation smart sensing technology is essential for bridging this gap. The societal challenge extends beyond technical capability to accessibility—how can cities with constrained budgets implement the advanced sensing infrastructure necessary for intelligent urban management? This research addresses this challenge by demonstrating that professional-grade structural monitoring can be achieved through the convergence of FBG sensing technology with open-source software architectures, making smart city infrastructures accessible to all municipalities regardless of economic constraints.

In South Africa, where 80% of Johannesburg's bridges are classified as being in poor condition [6], the implementation of accessible and cost-effective monitoring systems has become imperative for ensuring public safety and optimising maintenance resources. Traditional infrastructure monitoring approaches, characterised by periodic visual inspections at five-year intervals, have proven inadequate for detecting progressive structural deterioration [5]. The collapse of the bridge over the Ezimbokodweni River in the south of Durban, occurring merely eight months after receiving a satisfactory inspection report, exemplifies the critical limitations of qualitative assessment methods [5]. These conventional approaches cannot accurately determine the extent of fatigue damage, identify specific deterioration causes, or provide the quantitative measurements essential for data-driven maintenance decisions [7].

The evolution from first-generation electrical strain sensors to second-generation fibre optic sensor (FOS) technology represents a paradigm shift in monitoring capabilities [2]. First-generation sensors, based on electrical principles, featured short gauge lengths and provided a basis for local material monitoring. The second generation, particularly fibre Bragg grating (FBG) sensors developed in the 1970s and commercialised in the 1990s, greatly extended the applicability and effectiveness of strain-based SHM by providing long gauge and one-dimensional distributed sensing capabilities [2]. These sensors offer unique advantages, including immunity to electromagnetic interference, operational capability at extreme temperatures

ranging from -270°C to 1000°C, lightweight construction at approximately 150μm diameter, and intrinsic multiplexing capabilities enabling multiple sensors along a single fibre [8].

The wavelength-encoded operation of FBG sensors inherently provides immunity to intensity-related limitations that compromise other optical sensor types, making them particularly well-suited for long-term structural monitoring applications [9]. As noted by Glisic [2], the availability of long-gauge strain sensors was the first paradigm-changer brought about by fibre-optic strain sensor technologies, enabling monitoring at a global structural scale where the entire structure could be fitted with sensors, providing excellent spatial coverage.

The integration of FOS technology with modern digital infrastructure represents a convergence of physical sensing and cyber-physical systems, enabling the creation of digital twins that provide real-time virtual representations of physical structures [4]. This digital transformation aligns with Industry 4.0 principles, ISO/IEC/IEEE 42020:2019 architecture processes [10], and contributes directly to UN Sustainable Development Goal 9, which emphasises the development of resilient infrastructure, promotion of inclusive and sustainable industrialisation, and fostering of innovation [11].

Recent optical fibre sensing advances have achieved significant monitoring capabilities whilst revealing deployment barriers limiting widespread adoption. Applied quasi-distributed FBG sensors successfully monitor bridge structures through wavelength multiplexing and electromagnetic immunity, yet remain constrained by complex data acquisition [12]. Railway monitoring using distributed fibre optic sensing achieved three-dimensional deformation calculation and earthquake safety confirmation, but necessitates specially installed fibres and complex processing algorithms [13]. Tunnel monitoring with distributed optical fibre sensors effectively detected construction-induced structural changes, though suffering from bonding challenges and installation discontinuities [14]. Converting telecommunication fibre-optic cables into distributed acoustic sensors demonstrated cost-effective natural frequency identification, yet introduced substantial measurement noise requiring physics-guided analytical methods [15]. These implementations remain largely confined to research or specific field environments due to complexity, specialised installation requirements, and lack of accessible interfaces for practical deployment. The comprehensive digital twin system presented addresses these limitations through open-source web technologies, edge computing infrastructure, and standardised interfaces, enabling resource-constrained municipalities to deploy professional-grade monitoring.

This paper demonstrates evolution from traditional monitoring to accessible digital twin implementation, featuring 10 FBG sensors processing 1.72 million daily data points through WebSocket connections and Cloudflare edge computing across 24,869 lines of code. The architecture encompasses sensor digitisation following IEEE 2067-2021 and IEEE 1451.0-2024 standards, a comprehensive web dashboard with progressive app capabilities, automated alerts, and role-based access control. The main contribution is democratising professional-grade FBG sensing through open-source technologies, transforming complex research-confined systems into accessible solutions deployable via standard web browsers, thereby enabling widespread smart infrastructure monitoring without proprietary software licenses or specialised installation teams.

II. HISTORICAL EVOLUTION OF STRAIN SENSING TECHNOLOGIES

The development of strain-sensing technologies represents over a century of scientific advancement, beginning with Lord Kelvin's pioneering work on piezoresistive effects in 1856. This section traces the technological evolution from first-generation electrical strain gauges to modern fibre optic sensing systems, establishing the foundation for contemporary structural health monitoring applications.

A. From Electrical to Optical: A Century of Innovation

The journey of strain sensing in civil structures spans over 100 years of technological progress, with transformative milestones that have shaped modern SHM practices [2]. The theoretical foundation was established in 1856 when Lord Kelvin demonstrated that electrical resistance in metallic conductors changes under mechanical strain, sparking the first ideas about resistive strain gauges. However, practical implementation remained elusive due to the absence of suitable bonding agents until 1938, when the resistive strain gauge was independently invented by Edward E. Simmons and Arthur C. Ruge [2].

The first-generation electrical strain sensors, including vibrating wire (VW) sensors developed by Carlson in 1939, opened doors to understanding true structural behaviours in real-life settings and gave birth not only to strain-based SHM but to SHM in general [2]. These sensors featured high resolution, typically in the range of 1 με (1 microstrain) and precision ranging between 0.5% and 2% of full scale, making them suitable for civil structure monitoring [2].

B. The Fibre Optic Revolution

The second generation of strain sensors emerged with the development of fibre optic technologies in the latter half of the 20th century. Initial exploration of optical fibres for sensing purposes began almost immediately after Theodore Maiman's invention of the first laser in 1960 [2]. The physical principles behind fibre-optic sensors were developed by renowned scientists, including Albert A. Michelson, Charles Fabry, Alfred Perot, and others, resulting in two distinguished groups of sensors: discrete (point) sensors based on Extrinsic Fabry-Perot Interferometry (EFPI) and Fibre Bragg-Grating spectrometry (FBG), and truly distributed sensors based on Brillouin and Rayleigh scattering [2].

Discrete fibre optic strain sensors became available as both short-gauge sensors (up to 10 cm) and long-gauge sensors (250 mm and longer, up to 2 m), with commercial availability enabling the first widespread SHM applications in the 1990s [2]. The best performances of commercially available discrete fibre-optic strain sensor monitoring systems feature resolution typically in the range of 0.2–5 με and precision ranging between 1 με for discrete FOSS and Rayleigh-based distributed FOSS, and 20 με for Brillouin-based distributed FOSS [2].

C. Working Principle of Fibre Bragg-gratings (FBGs)

A fibre Bragg grating sensor essentially reflects a portion of the light launched to the FBG corresponding to the Bragg wavelength and transmits the rest of the spectrum[16]. Changes in temperature, strain, vibration, and other environmental parameters cause a shift in the Bragg wavelength.

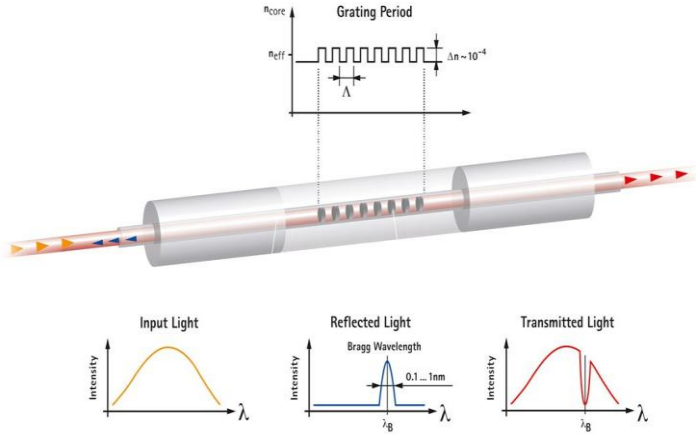


Figure 1: FBG operation as illustrated in [17].

Figure 1 demonstrates the operation of an FBG, in which a broadband light signal is launched into an optical fibre and a narrowband signal is reflected from the fibre Bragg grating, while the rest of the broadband signal's spectrum is transmitted.

The reflected Bragg wavelength is symbolised as λ_B. It is related to the core refractive index and the grating period of the index modulation, expressed as follows [18]:

$$\lambda_B = 2n_{eff}\Lambda \qquad (1)$$

where n_{eff} is the core refractive index and Λ is the grating period. As mentioned previously, strain, temperature, vibration or any environmental factor for that matter, cause changes to the grating period Λ. This will cause a shift in the Bragg wavelength λ_B, and this relation can be equated by a data acquisition system as follows [18]:

$$\frac{\Delta\lambda_B}{\lambda_B} = (1 - p_e)\varepsilon + (\propto_\Lambda + a_n)\Delta T \qquad (2)$$

Where $\Delta\lambda_B$ is the shift in the reflected wavelength, λ_B is the initial wavelength, p_e is the photoelastic coefficient, ε is the measured strain, $\propto_\Lambda$ is the thermal coefficient, a_n is the thermal modulation of the core refractive index, and ΔT is the measured temperature shift [19].

The sensors are manufactured by inscribing fibre Bragg gratings (FBGs). An FBG is a periodic change in the refractive index of the glass core of an optical fibre.

III. MONITORING SYSTEM ARCHITECTURE AND DIGITAL INFRASTRUCTURE

Modern structural health monitoring requires sophisticated digital architectures that can seamlessly integrate physical sensing with cyber-physical systems to create functional digital twins. This section presents the comprehensive system architecture following ISO/IEC/IEEE standards, detailing the sensor digitalisation process, edge computing deployment, and the technology stack enabling real-time global accessibility.

A. Architecture Process Framework

The system architecture follows the principles defined in ISO/IEC/IEEE 42020:2019, which establishes a coherent set of processes for governance, management, conceptualisation, evaluation, and elaboration of architectures [10].

B. Sensor Digitalisation and Industry Decentralisation

The transformation of physical phenomena into digital information forms the foundation of modern SHM systems. Figure 2 illustrates the comprehensive sensor digitalisation architecture that enables the capture, processing, and distribution of structural health data to remote stakeholders.

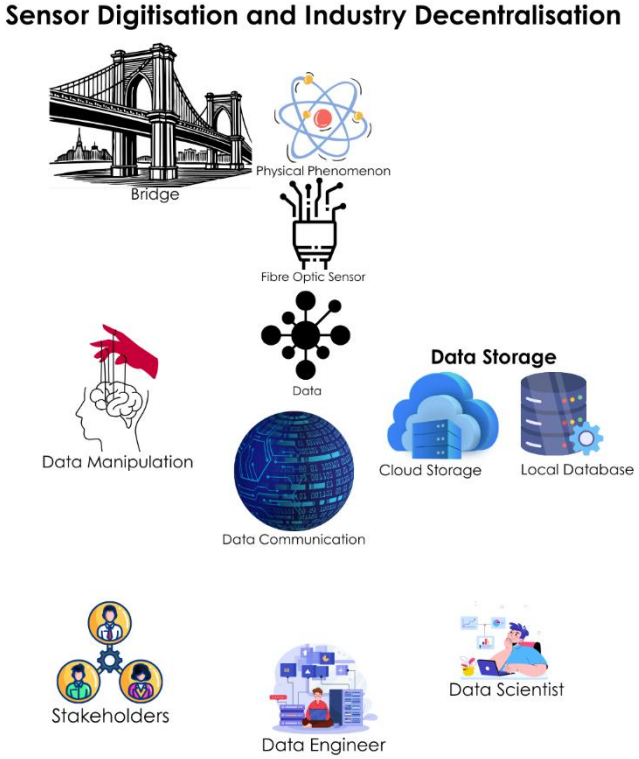


Figure 2: Sensor Digitalisation and Industry Decentralisation - showing the flow from physical sensors through digital conversion, edge processing, cloud infrastructure, to remote clients and stakeholders.

The digitisation process encompasses four critical stages: (1) optical signal generation and acquisition through FBG sensors responding to structural strain and temperature changes, (2) wavelength interrogation using the BraggMETER FS22SI with 10^{-4} nm sensitivity, (3) digital conversion via SCPI protocol commands over TCP/IP following IEEE 1451.0-2024 standards [20], and (4) data structuring into JSON format for transmission. This process enables the transformation of wavelength shifts, typically in the picometer range, into multiple useful metrics, including strain and temperature values. The process also enables the real-time relay of such information through a web dashboard.

C. Smart Transducer Interface Standards

The implementation adheres to IEEE 1451.0-2024, the Standard for Smart Transducer Interface for Sensors and Actuators, which provides common functions, communication protocols, and Transducer Electronic Data Sheet (TEDS) formats [20]. This standard enables plug-and-play integration and enhances flexibility across diverse IoT devices by defining a common interface that enables seamless communication

between various devices from different manufacturers [20]. The TEDS capability allows transducers to self-identify, providing information about their characteristics and ensuring compatibility across the monitoring system.

D. Digital Technology Stack

The implementation leverages a comprehensive technology stack comprising multiple programming languages, as detailed in Table 1.

Table 1: Programming Language Usage

Language	Files	Purpose
TypeScript JSX	52	Frontend UI components
Python	17	Backend processing
TypeScript	12	Type definitions
JSON	10	Configuration
PostCSS	1	Styling
JavaScript	3	Utilities
Total	**99**	**Complete System**

The technology stack selection prioritises performance, maintainability, and scalability. Python's asyncio capabilities enable non-blocking I/O operations essential for real-time data acquisition, while TypeScript provides type safety crucial for maintaining large-scale frontend applications. The substantial JSON configuration ensures system flexibility and enables runtime parameter adjustments without code modifications.

E. Edge Computing Infrastructure

The deployment utilises Cloudflare's global edge network to minimise latency and enhance reliability. Figure 3 demonstrates the edge computing architecture that enables secure, low-latency data transmission from the local monitoring site to global users.

The Cloudflare tunnel bypasses traditional firewall restrictions while maintaining security through encrypted connections. This architecture achieves several critical objectives: (1) elimination of public IP requirements for the local server, (2) automatic DDoS protection through Cloudflare's infrastructure, (3) global content delivery network (CDN) distribution, reducing latency to under 50ms for most geographical locations, and (4) WebSocket protocol support enabling real-time bidirectional communication.

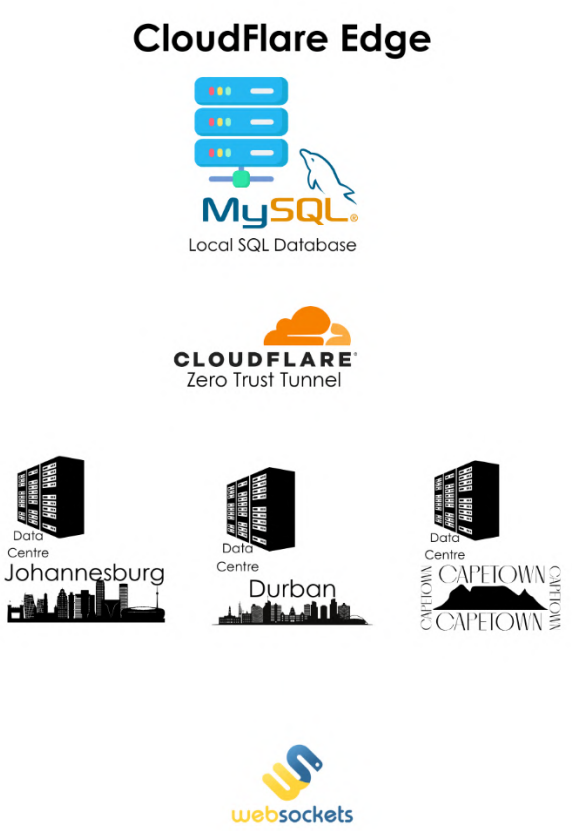


Figure 3: Cloudflare Edge - tunnel connection allows local database to transmit data through edge datacentres to WebSocket endpoints

IV. DATA ACQUISITION AND PROCESSING

The transformation of optical signals into informative engineering data requires precise calibration, robust data processing pipelines, and efficient management of substantial data volumes. This section examines the FBG sensor manufacturing and deployment methodology, real-time WebSocket-based data distribution, and the big data strategies managing 1.72 million daily measurements.

A. FBG Sensor Manufacturing and Calibration

The FBG sensors were manufactured in-house using the phase mask technique with the 266nm harmonic of an Nd:YAG laser. Ten sensors were fabricated with carefully selected wavelengths distributed across the 1534-1562nm range to prevent spectral overlap during the multiplexing operation.

Calibration experiments confirmed the theoretical sensitivities: strain sensitivity of 1.08 pm/με and temperature sensitivity of 10.6 pm/°C. These values align with the fundamental FBG response equation (2), validating the manufacturing quality and ensuring measurement accuracy[18].

B. Bridge Monitoring Setup

The B2LAB-B3LAB pedestrian bridge served as the testbed for the system validation. Figure 4 illustrates the sensor deployment configuration across the 10-metre span structure.

The installation employs a dual-channel configuration: Channel 0 contains five temperature compensation sensors

housed in aluminium tubes to isolate mechanical strain whilst maintaining thermal coupling. Channel 1 features five strain measurement sensors directly bonded to the bridge structure using strain gauge epoxy, ensuring optimal strain transfer from the concrete substrate to the optical fibre.

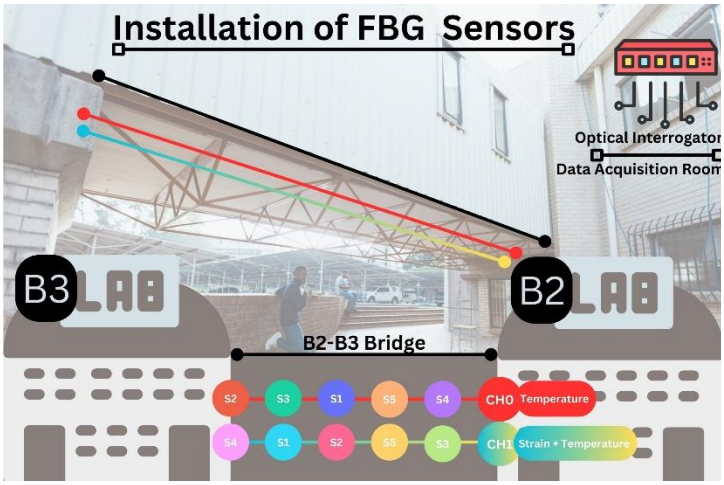


Figure 4: B2LAB-B3LAB Bridge Monitoring Setup - showing sensor positions and channel allocation

C. Real-time Data Processing Pipeline

The data acquisition system implements a pipeline transforming raw optical measurements into engineering parameters. Figure 5 illustrates the WebSocket-based real-time distribution architecture.

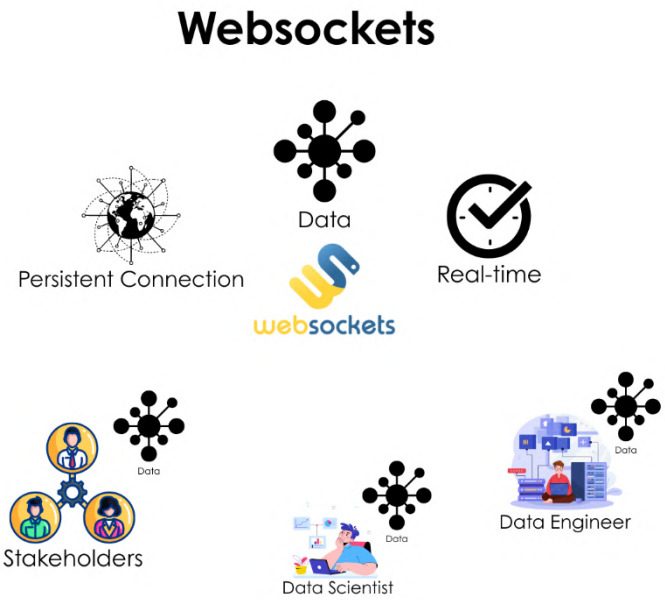


Figure 5: WebSocket Data Distribution - showing persistent connections and Data Delivery, in real-time

The WebSocket implementation achieves: (1) persistent bidirectional connections eliminating polling overhead, (2) automatic reconnection with exponential backoff preventing server overload, (3) message compression reducing bandwidth requirements by approximately 70%, and (4) multi-client synchronisation ensuring all connected dashboards display identical real-time data.

D. Big Data Management

The continuous 1Hz sampling rate generates substantial data volumes requiring efficient management strategies. Figure 6 presents the daily data collection metrics for the deployed system.

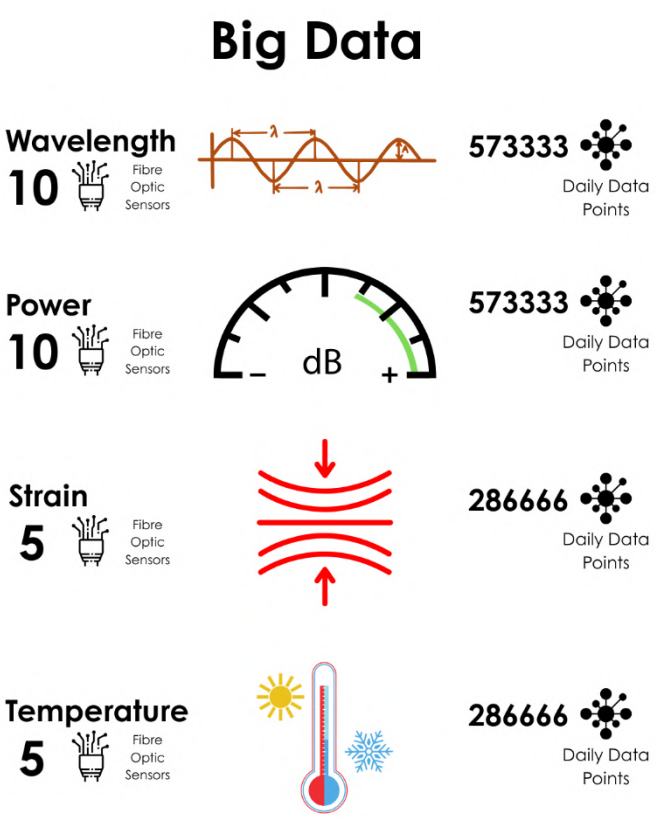


Figure 6: Big Data Collection Metrics - showing the amount of readings/day per sensor type (1.72M total data points/day).

Daily Data Volume Calculations:

- Recording rate: 0.67Hz (1 sample/1.5second)
- Daily samples per sensor: 57333 (24h × 3600s)
- Total FBG sensors: 10 (2 channels × 5 sensors)
- Metrics per reading: 4 (wavelength, power, strain, temperature)
- **Total daily data points: 1,719,999**

The system implements intelligent data partitioning with daily SQLite database files, preventing single-point corruption risks whilst enabling efficient historical queries. Firebase Firestore provides cloud backup with a 19,200 writes/day quota managed through batch upload optimisation.

V. SOFTWARE ARCHITECTURE

The implementation of a professional-grade monitoring platform demands careful consideration of both backend processing capabilities and frontend user experience to ensure accessibility across diverse stakeholder groups. This section details the microservices architecture, progressive web application development, and comprehensive security

measures that enable reliable remote monitoring through standard web technologies.

A. Backend Architecture

The backend system, accessible at api.masterfilmon.com, implements a microservices architecture using FastAPI 0.78.0 with Uvicorn ASGI server. Figure 7 details the comprehensive backend architecture enabling distributed data processing and frontend gateway through shm.masterfilmon.com.

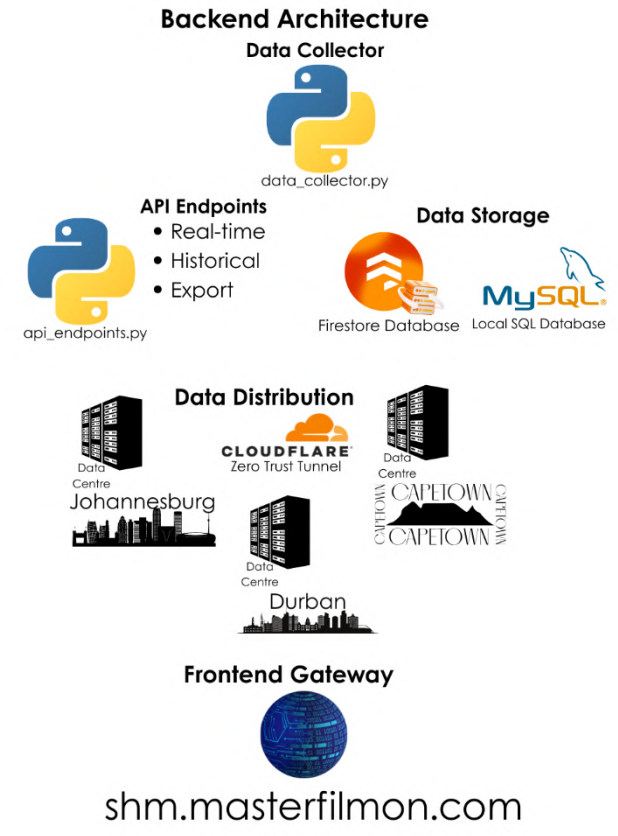


Figure 7: Backend Architecture - Real-time sensor monitoring system with distributed data processing and shm.masterfilmon.com as a frontend gateway.

Key architectural components include:

1. **Data Collection Layer**: SCPI protocol is used to communicate with the BraggMETER device at IP address 10.0.0.10 and port 3500. The BraggMETER generates data at 1Hz.
2. **Processing Pipeline**: Asynchronous Python co-routines handle concurrent operations such as:
    - Wavelength to strain conversion.
    - Temperature compensation.
    - Local and cloud data storage.
3. **API Endpoints**: RESTful architecture with OpenAPI 3.0 specification provides the following functionality:
    - /api/realtime: WebSocket endpoint for live data streaming.
    - /api/historical: Time-series data retrieval with pagination.
    - /api/export: Data export in CSV/JSON formats.
    - /api/health: System diagnostics and monitoring.
4. **Security Layer**: Multi-layered security implementation ensures data safety through:
    - Firebase Authentication integration.
    - JWT token validation with automatic refresh.
    - CORS configuration for cross-origin requests.
    - Rate limiting preventing API abuse.

B. Frontend Architecture

The frontend dashboard, deployed at shm.masterfilmon.com, utilises Next.js 15.5.2 with React 18.3.1 and TypeScript. Figure 8 presents the modern web dashboard enabling responsive, real-time visualisation.

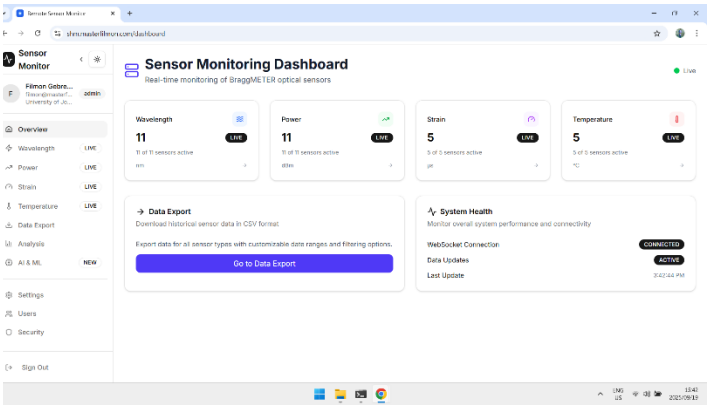


Figure 8: Frontend Dashboard - Real-time sensor monitoring dashboard with modern web technologies.

Figure 9 shows the live monitoring graph for strain sensors (wavelength, power and temperature have the same functionality).

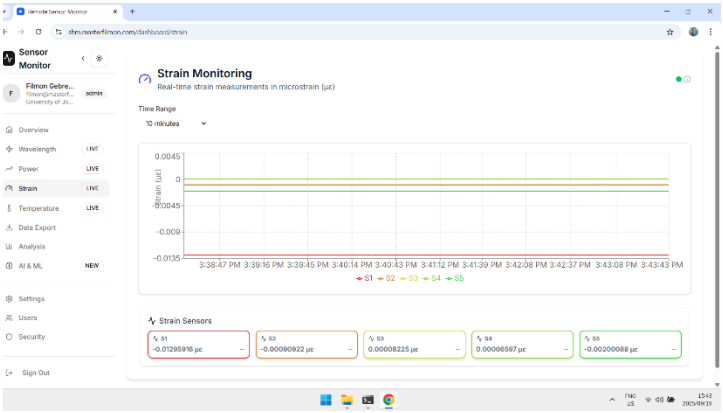


Figure 9: Strain Monitoring: live WebSocket data at the bottom of the dashboard and real-time graph.

Figure 10 shows the Data Export page, where data can be downloaded as a cleaned CSV file. Date range, metric type and sensor set for data download are selectable.

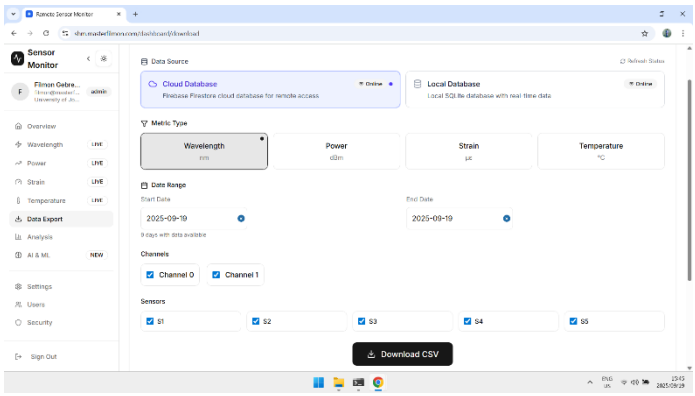


Figure 10: Logged-in users have access to recorded data that can be exported as CSV files from the cloud Firestore database.

The implementation features:

1. **Component Architecture**: 52 TypeScript JSX files implementing modular, reusable components with React hooks for state management and TanStack Query for server synchronisation.
2. **Real-time Visualisation**: Recharts library rendering time-series data with:
    - 1Hz update rate.
    - Zoom and pan capabilities for detailed analysis.
    - Multi-metric overlay displaying strain, temperature, and derived parameters.
    - Responsive design adapting to viewport sizes from 320px to 1920px+.
3. **Progressive Web App**: Service worker implementation enabling:
    - Offline functionality with data caching.
    - Push notifications for alerts via WhatsApp (See Figure 11, Figure 12 and Figure 13).
    - Installation as native application.

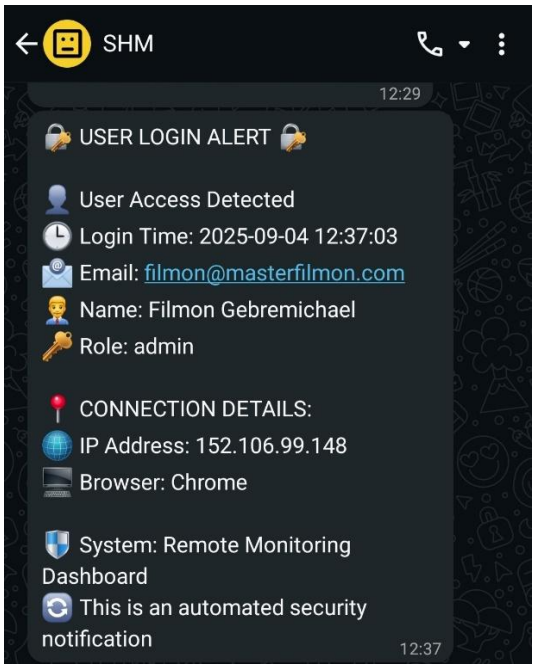


Figure 11: Automated WhatsApp User Login Alert.

Figure 12 shows a snapshot of database connections and all current sensor readings.

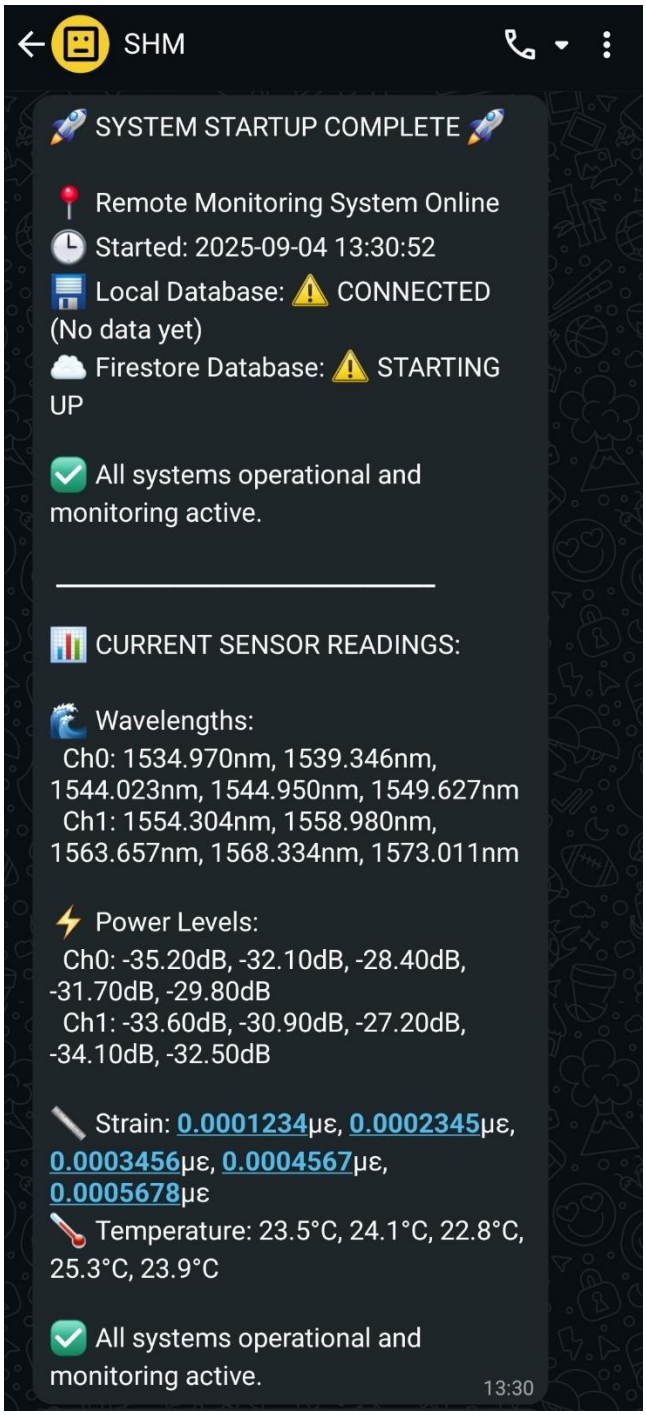


Figure 12: System startup WhatsApp Alert indicating database connections and current sensor readings.

The same occurs when the system is disconnected. And then again, when the system is reconnected, an alert is sent out. Figure 13 shows the disconnection alert.

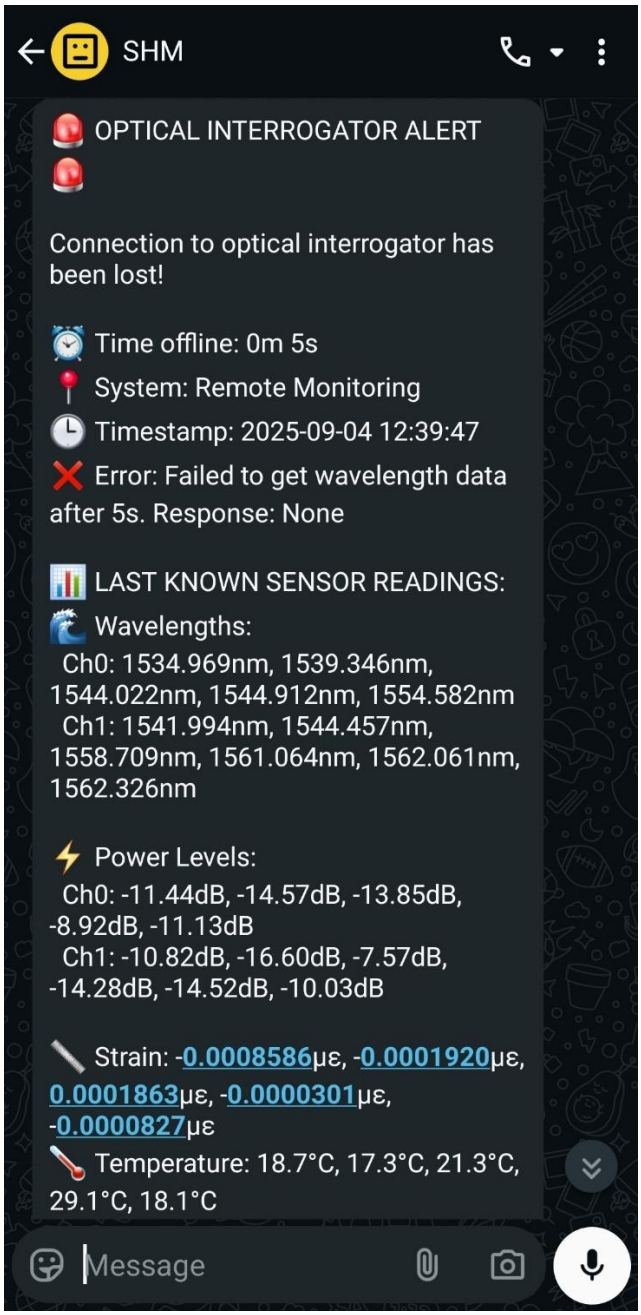


Figure 13: Disconnection WhatsApp Alert indicating current sensor readings and disconnection time.

The details within these alerts can be programmatically set. Meaning the user can decide which parameters are relevant to be alerted on.

C. User Authentication and Security

The system implements comprehensive security measures, ensuring data integrity and access control. Figure 14 illustrates the multi-factor authentication flow.



Figure 14: User Authentication - Firebase enables role-based access control, admin tools and security measures like TLS.

Authentication features include:

- Single Sign-On (SSO) via Firebase Authentication.
- Role-based access control (admin, engineer, operator, viewer). Users are registered in an authentication database with their email, password, and role.
- Session management with automatic timeout.
- Encrypted data transmission using TLS 1.3.

VI. DISCUSSION

The successful implementation validates several key hypotheses regarding modern SHM system design. The integration of FBG sensors with contemporary web technologies demonstrates that research-grade monitoring capabilities can be delivered through accessible, cost-effective platforms. This achievement represents the culmination of over 100 years of strain sensing evolution, from Lord Kelvin's initial observations to today's sophisticated digital twin implementations [2].

The choice of open-source technologies and cloud infrastructure significantly reduces deployment barriers whilst maintaining professional-grade performance. The WebSocket-based real-time architecture proves particularly effective for structural monitoring applications, eliminating the latency and overhead associated with traditional polling approaches [21], [22]. As documented in RFC 6455, polling-based approaches suffer from high protocol overhead and multiple TCP connections per client [21]. Pimentel and Nickerson demonstrated that WebSocket reduces latency by 50-500ms compared to HTTP polling in real-time sensor applications [22]. The persistent connection model ensures that critical events are immediately propagated to all connected clients, enabling rapid response to anomalous conditions.

The implementation of IEEE 1451.0-2024 standards provides crucial interoperability benefits, enabling the system to potentially integrate with diverse sensor types and communication protocols [11]. The TEDS capability allows for self-identification and configuration of sensors, reducing setup complexity and enabling true plug-and-play functionality. This standardisation is particularly important for future system expansion and integration with emerging IoT technologies.

The edge computing deployment through Cloudflare infrastructure addresses several traditional challenges in remote monitoring systems. By eliminating the need for static IP addresses and complex firewall configurations, the system can be deployed in diverse environments without extensive IT infrastructure modifications. The global CDN ensures consistent performance regardless of user location, making the system suitable for international collaboration and remote expertise consultation.

From a smart cities' perspective, this implementation addresses critical urban challenges identified in the literature [3], [4]: ageing infrastructure requiring cost-effective monitoring, the need for data-driven maintenance decisions to optimise limited municipal resources, and the imperative for real-time safety assurance in public spaces. The system's accessibility via standard web browsers and mobile devices enables city engineers to monitor infrastructure from anywhere, supporting rapid emergency response—a key requirement for smart city resilience [4]. The digital twin capability aligns with Industry 4.0 principles [4] and smart city frameworks that emphasise cyber-physical system integration. The open-source approach particularly benefits developing smart cities, where budget constraints often prevent the adoption of commercial monitoring solutions.

VII. CONCLUSION

This research contributes to the smart cities paradigm by demonstrating that advanced FBG sensing networks, essential for intelligent urban infrastructure [3], can be transformed into accessible monitoring tools through modern digital architectures.

Utilising current digital technology stacks and well-established fibre optic sensors (FOS), it is possible to design and implement digital twins that enable SHM. The data pipeline described in this text is end-to-end. Raw optical data travels from the physical point of origin all the way to a client (stakeholder, data engineer, data scientist, subject-matter expert) in a remote location to provide real-time information.

The code will be expanded in the future. The expansion will be based on future calibration experiments, new standards for structural health monitoring, as well as advances in AI, ML, statistics, and programming in general. It can also be expanded by a bigger sensor set or a larger real-world application.

This research demonstrates that the convergence of optical sensing technology with modern software engineering practices enables the creation of sophisticated monitoring systems that are both technically advanced and practically accessible. Building upon a century of strain-sensing evolution, the implementation showcases how historical innovations converge with contemporary digital technologies.

The successful deployment on the B2LAB-B3LAB bridge validates the feasibility of implementing research-grade SHM systems using commodity hardware and open-source software, significantly reducing the traditional barriers to adoption. The system's adherence to international standards, including IEEE 1451.0-2024 for smart transducers, IEEE 2067-2021 for FBG interrogators, and ISO/IEC/IEEE 42020:2019 for architecture processes, ensures interoperability and future-proofing of the implementation.

The system's contribution to UN SDG 9 extends beyond technical implementation. By providing open documentation, modular architecture, and standard-compliant interfaces, the research enables technology transfer and capacity building essential for sustainable infrastructure development. The demonstrated open-architecture software and accessibility over any internet-connected device make the technology applicable to resource-constrained environments where traditional monitoring systems remain economically unfeasible.

The real-time digital twin capability transforms infrastructure management from reactive to proactive, enabling data-driven decision-making that optimises resource allocation, extends asset lifespan, and enhances public safety. As infrastructure worldwide faces increasing demands from urbanisation, climate change, and ageing, the technologies and methodologies presented in this research provide a pathway toward more resilient and sustainable infrastructure systems.

ACKNOWLEDGMENT

This work has been supported by Telkom, the National Laser Centre (NLC) and the Photonics Research Centre at the University of Johannesburg.

REFERENCES

[1] C. R. Farrar and K. Worden, “An introduction to structural health monitoring,” 2006, doi: 10.1098/rsta.2006.1928.

[2] B. Glisic, “Concise Historic Overview of Strain Sensors Used in the Monitoring of Civil Structures: The First One Hundred Years,” *Sensors*, vol. 22, no. 6, p. 2397, Mar. 2022, doi: 10.3390/s22062397.

[3] S. Sony, S. Laventure, and A. Sadhu, “A literature review of next-generation smart sensing technology in structural health monitoring,” *Struct Control Health Monit*, vol. 26, p. 2321, 2019, doi: 10.1002/stc.2321.

[4] Y. Li and Y. Zhang, “Digital twin for Industrial Internet,” *Fundamental Research*, vol. 4, no. 1, pp. 21–24, Jan. 2024, doi: 10.1016/j.fmre.2023.01.005.

[5] eNCA, “SA’s broken infrastructure | Bridge collapse sparks concern,” *eNCA*, Nov. 08, 2024. Accessed: Jan. 01, 2025. [Online]. Available: https://www.enca.com/videos/sas-broken-infrastructure-bridge-collapse-sparks-concern

[6] eNCA, “SA’S BROKEN INFRASTRUCTURE | ALMOST 80% OF JOBURG’S BRIDGES AT RISK OF COLLAPSE,” *eNCA*, Oct. 25, 2024. Accessed: Jan. 01, 2025. [Online]. Available: https://www.enca.com/business-videos/discussion-sas-broken-infrastructure-concern-over-structural-integrity-bridges#

[7] K. Worden, C. R. Farrar, G. Manson, and G. Park, “The fundamental axioms of structural health monitoring,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 463, no. 2082, pp. 1639–1664, Jun. 2007, doi: 10.1098/rspa.2007.1834.

[8] X. W. Ye, Y. H. Su, and J. P. Han, “Structural Health Monitoring of Civil Infrastructure Using Optical Fiber Sensing Technology: A Comprehensive Review,” 2014, doi: 10.1155/2014/652329.

[9] T. H. T. Chan *et al.*, “Fiber Bragg grating sensors for structural health monitoring of Tsing Ma bridge: Background and experimental observation,” *Eng Struct*, vol. 28, no. 5, pp. 648–659, Apr. 2006, doi: 10.1016/j.engstruct.2005.09.018.

[10] “ISO/IEC/IEEE International Standard - Software, systems and enterprise -- Architecture processes,” May 21, 2019, *IEEE, Piscataway, NJ, USA*. doi: 10.1109/IEEESTD.2019.8767004.

[11] United Nations, “Transforming our world: the 2030 Agenda for Sustainable Development ,” 2015.

[12] P. Banda, M. Della Tamin, and J. Meyer, “Applied Quasi-Distributed Fibre Bragg Grating Strain Sensors in Structural Damage Sensing of a Bridge Structure,” *SAIEE Africa Research Journal*, vol. 113, no. 2, pp. 75–87, Jun. 2022, doi: 10.23919/SAIEE.2022.9785541.

[13] K. Kishida, T. L. Aung, and R. Lin, “Monitoring a Railway Bridge with Distributed Fiber Optic Sensing Using Specially Installed Fibers,” *Sensors*, vol. 25, no. 1, p. 98, Dec. 2024, doi: 10.3390/s25010098.

[14] J. Gómez, J. R. Casas, and S. Villalba, “Structural Health Monitoring with Distributed Optical Fiber Sensors of tunnel lining affected by nearby construction activity,” *Autom Constr*, vol. 117, p. 103261, Sep. 2020, doi: 10.1016/j.autcon.2020.103261.

[15] J. Liu, S. Yuan, B. Luo, B. Biondi, and H. Y. Noh, “Turning Telecommunication Fiber-Optic Cables into Distributed Acoustic Sensors for Vibration-Based Bridge Health Monitoring,” *Struct Control Health Monit*, vol. 2023, pp. 1–14, Apr. 2023, doi: 10.1155/2023/3902306.

[16] L. Schenato, “Fiber-Optic Sensors for Geo-Hydrological Applications: basic concepts and applications,” in *Rendiconti online della Società Geologica Italiana*, Roma: Società Geologica Italiana, 2014, pp. 51–54.

[17] C. Barbosa, “Optical Fiber Sensors vs. Conventional Electrical Strain Gauges for Infrastructure Monitoring Applications,” Boston.

[18] J. Yumba, M. Grobler, and M. Ferentinou, “Using Fibre Bragg grating sensors for monitoring small-scale sinkhole occurrences: an experimental study,” in *Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2022*, M. Smuts and R. Moorcroft, Eds., 2022.

[19] J. Yumba, “Laboratory investigation of sinkhole propagation induced by a leaking pipe using fibre Bragg grating sensors,” Dissertation, University of Johannesburg, Johannesburg, 2022.

[20] “IEEE Standard for a Smart Transducer Interface for Sensors and Actuators--Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats,” Feb. 15, 2024, *IEEE, Piscataway, NJ, USA*. doi: 10.1109/IEEESTD.2024.10571828.

[21] I. Fette and A. Melnikov, “The WebSocket Protocol,” Dec. 2011. doi: 10.17487/rfc6455.

[22] V. Pimentel and B. G. Nickerson, “Communicating and Displaying Real-Time Data with WebSocket,” *IEEE Internet Comput*, vol. 16, no. 4, pp. 45–53, Jul. 2012, doi: 10.1109/MIC.2012.64.

Digitization of Analogue Gauges using Computer Vision

1st Erico Kayumba Mutonkole
dept. of Computer Science
University of the Western Cape
Cape Town, South Africa
3805086@myuwc.ac.za

2nd Olasupo Ajayi
dept. of Computer Science
University of the Western Cape
Cape Town, South Africa
ooajayi@uwc.ac.za

3rd Omowunmi Isafiade
dept. of Computer Science
University of the Western Cape
Cape Town, South Africa
oisafiade@uwc.ac.za

***Abstract*—Analogue gauges remain prevalent in manufacturing industries for measuring physical quantities, yet manual monitoring by operators introduces human error and operational inefficiency. This research developed and evaluated an automated gauge reading system using computer vision and deep learning techniques. Three hybrid approaches were implemented and compared: OpenCV with Scikit-Image (traditional computer vision), OpenCV with ResNet18, and OpenCV with VGG19 (deep learning approaches). A comprehensive dataset of 1,848 pressure gauge images (0-10 BAR range) was curated through web scraping, synthetic generation, and augmentation techniques. The experimental results demonstrate that ResNet18 significantly outperforms the alternatives, achieving an accuracy of 97. 3%, compared to VGG19's 64.4% and Scikit-Image's 63. 8%. The study validates the effectiveness of residual learning architectures for industrial gauge digitization, providing a robust foundation for Industry 5.0 automation applications.**

***Index Terms*—Analogue Gauge, Computer Vision, Image Processing, OpenCV, Pytorch, ResNET18, Scikit-Image, VGG19.**

I. Introduction

Analogue gauges are scientific instruments designed to measure physical quantities. These include temperature, mass and voltage to name a few. Despite being an older technology, analogue gauges are still widely used in many industries today to take measurement readings for various equipment [4]. However, it is easier for users to read digital numbers compared to reading from an analogue gauge thereby, reducing guesses of what value the meter pointer is pointing to [2].

Manual gauge reading in industrial environments presents significant operational challenges and costs. Studies indicate that manual readings can have error rates of up to 5%, often due to fatigue, distraction, or misinterpretations [2], [12], while broader research on manual data entry shows typical error rates of about 1%, with some studies reporting rates as high as 4% for data entered without verification layers [13], [14]. In large industrial plants such as oil and gas refineries, manual reading tasks are very error-prone, time-consuming and often dangerous due to exposed positions of the gauges [5]. These errors translate into production inefficiencies, unnecessary process adjustments, and potential safety risks in critical industrial operations.

When we take cost into consideration, especially in industrial settings, rather than installing new digital meters, it might be cheaper to attach tiny cameras attached to cheap single board computers, such as Raspberry PI or ESP32, to capture images that will be processed by image recognition models, to obtain numeric readings [2]. Despite their practicality, a comprehensive comparison between traditional computer vision approaches and modern deep learning architectures for gauge digitization on low resource devices remains limited in existing literature.

This work aims at developing a system that converts analogue readings to their digital equivalent using computer vision. The research also addresses the gap in literature by systematically comparing three distinct approaches: traditional computer vision (OpenCV with Scikit-Image), modern residual learning (OpenCV with ResNet18), and deep sequential learning (OpenCV with VGG19). The primary research objective is to determine whether architectural innovations in deep learning, particularly residual connections, provide significant performance advantages over traditional methods for analogue gauge classification tasks. The OpenCV library will be used to preprocess images of analogue gauges, which will then be used to train the deep learning model. The trained model will accurately determine the numerical or digital equivalent value of the gauge's reading.

II. LITERATURE REVIEW

A recent review on deep learning approaches for digital image processing [1] examined a wide range of techniques tailored for various application areas. The study provided a thorough investigation into several image processing tasks, including noise reduction, image enhancement, segmentation, feature detection, and classification. To evaluate the performance and effectiveness of these methods, several metrics were employed, such as mean squared error (MSE), peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), mean structural similarity index (MSSIM), mean absolute error (MAE), and naturalness image quality evaluator (NIQE), among others. The review highlighted the strong noise reduction capabilities of methods like Self2Self neural networks, DnCNNs, and DFT-Net. In terms of classification, architectures ranging from Residual Networks to combinations like CNN-LSTM were recognized for their promising accuracy [1].

In a study focused on water meter reading recognition using deep learning techniques [2], three models namely Faster R-

CNN, SSD, and YOLOv3 were evaluated to identify which offered superior performance. Among the models assessed, YOLOv3 achieved the highest recognition accuracy, recording a success rate of 90.61%.

A proposed automated system combined deep learning, machine learning, and image processing to detect readings from analogue gauges [3]. The researchers utilized image processing techniques to generate supervised data from gauge images, addressing the challenges of working with unsupervised data and aiming to improve recognition accuracy. Three models—DenseNet 169, InceptionNet V3, and VGG19—were trained on 1,011 labeled images categorized into 9 classes with labels from 0 to 8. VGG19 showed a high training accuracy of 97% but a lower testing accuracy of 75%, indicating possible overfitting. In contrast, InceptionNet V3 showed consistent performance across both datasets, while DenseNet 169 delivered the highest accuracy and strongest generalization ability among the models [3].

In an effort to automate analogue gauge reading across various industries, including healthcare and automotive, researchers in [4] developed an algorithm called the Analogue Gauge Reader (AGR) using artificial intelligence methods. The algorithm was implemented with open-source tools such as OpenCV, NumPy, and PIL. Experimental results demonstrated a notable reduction in error rates following the application of image preprocessing techniques. Moreover, the algorithm proved adaptable to a range of analogue gauge types captured under varying lighting conditions. In testing, the approach achieved high accuracy and precision, ranging from 99% to 100% [4].

Analogue meters located in areas that were hard-to-access or hazardous locations proved to be challenging as measurements for various equipment in the manufacturing plant were needed. A solution was proposed in [5] to remotely monitor these analogue gauges in areas that are difficult to reach or hazardous using a combination of a microcontroller device and a camera. The process involved using the camera to capture gauge images, which were then processed on a local computer using a Convolutional Neural Network (CNN) with the CenterNet HourGlass104 model, segmentation of the pointer was done with the Hough Transform, while polar transformation was used to calculate the pointer angle. The value indicated by the gauge was determined based on the pointer's angle, and displayed on a web-accessible dashboard. The approach achieved an average relative error of 0.95%, demonstrating good accuracy and robustness for industrial environments [5].

A real-time mobile solution for reading circular single-pointer analogue gauges using a lightweight CNN-based system was presented in [6]. They used a synthetic dataset of 10,000 images with ground truth annotations to train the CNN model they implemented. The focus of the research was to implement an efficient method for processing both high- and low-resolution gauge images on mobile phones, and to the release of a diverse real-world gauge dataset. They achieved pointer angle errors below 1° and while running at up to 25fps on mobile devices [6].

III. RESEARCH FOCUS

This work addresses the fundamental challenge of automated interpretation of analogue gauges using computer vision. As a proof of concept and to demonstrate the viability of this idea, only pressure gauges of up to 10 BAR were considered. The research investigates three distinct hybrid approaches: OpenCV with Scikit-Image (traditional computer vision), OpenCV with ResNet18 (modern residual learning), and OpenCV with VGG19 (deep sequential architecture). This comparative framework enables systematic evaluation of traditional versus deep learning methodologies while examining the impact of architectural innovations (residual connections) on classification performance. The choice of these approaches is grounded in their theoretical significance. Scikit-Image represents traditional computer vision techniques, VGG19 illustrates deep sequential architectures that are susceptible to vanishing gradient issues [18], and ResNet18 demonstrates modern advancements designed to overcome gradient flow challenges. Their performance is evaluated using accuracy, precision, recall, F1-score and mean average precision (mAP), providing a comprehensive assessment across multiple dimensions.

IV. REQUIREMENT ANALYSIS

Computer vision has vast industrial applications and has steadily been growing in popularity in recent times. This work's scope is limited to image recognition, image processing and deep learning models and their applications in digitizing analogue gauges.

To be considered practical and viable, the proposed solution must meet the following functional requirements: Images of analogue gauge must be acquired from camera sensors.

1) Computer vision models should be used to analyse the captured image to obtain the equivalent digital reading.
2) Digital value inference must have a precision targeting ±0.1 BAR accuracy
3) Real-time processing capability for industrial applications, with approximately 2 seconds processing time per image.
4) The developed system should be able to work on low-spec hardware such as microcontrollers or single board computers

V. METHODOLOGY AND EXPERIMENTAL DESIGN

Analogue gauge digitization is formulated as a multi-class classification problem where each discrete gauge reading (0-10 BAR) represents a distinct class. This approach, while discretizing continuous readings, provides robust performance against visual noise and simplifies the mapping from visual input to numerical output compared to regression-based needle angle estimation methods [20]. The following sub-sections discuss our research methodology.

A. Data Collection and Preparation

The data for this experiment were prepared as follows:

1) A custom dataset was curated by web scraping pressure gauge images from Google Images. This was necessary because, for the task of analogue gauge reading, there is a significant lack of large, publicly available datasets, a limitation also acknowledged by [6]. Three (3) types of pressure gauges were selected as shown in Figure 1. These gauges had readings from 0 to 10 BAR (11 classes).
2) The 3 images were then synthesized to generate 33 image classes using Adobe Photoshop. 11 images per class * 3 = 33. A sample of the synthesized image is shown in Figure 2.
3) Finally, we augmented the resulting 33 images using various techniques and ended up with 1,848 augmented pressure gauge image variants.



Fig. 1. Three original gauges sourced from Google Images.



Fig. 2. Original (Left) and synthetic (right) image.

1) Data augmentation pipeline: The augmentation process was designed to improve model robustness to real-world variations in gauge imaging. Five augmentation processes were carried out.

1) The images were randomly rotated by ±15° to simulate different camera angles.
2) The images were scaled between 90% and 110% to account for changes in camera distance.
3) Positional robustness was introduced through translation of up to 10% of the image's width and height along both axes.
4) Perspective transformations were applied to mimic changes in viewing angles.
5) Salt-and-pepper noise injection was used to simulate sensor noise and environmental artifacts.



Fig. 3. Samples of augmented images.

2) Dataset Partitioning Pipeline: The dataset was then split into an 80:20 ratio. 80% for training (80%, n=1.474) and 20% for Validation (20%, n=374).

B. Image Processing

The following steps were taken to process the images in the dataset:

1) Conversion to Grayscale
   The images were converted to grayscale. Removing colour information simplifies the images and enhances the intensity. This step has been shown to help in reducing computational complexity and is essential for edge detection and further analysis, as reported in [7]. A comparison of the original vs gray scale version is shown in Figure 4
2) Extraction of Region of Interest (RoI)
   We isolated the gauge area from the background to focus on the circular dial of the gauge and filter out irrelevant parts of the image.
3) Gauge Labelling



Fig. 4. Original (left) image and Grayscale (right) image.

Traditionally, when using computer vision for analogue gauge digitization, the model estimates the needle's angle to determine the reading as documented in [21]. In contrast, this study used a classification-based approach, treating readings as discrete labeled values. This ensured that our computer vision models would be more robust to visual noise and simplify the mapping from image to value. To achieve this we framed this work as an image classification problem and utilized deep learning models (ResNet18 and VGG19). Being supervised learning models, we had to first manually label the images, assigning each gauge image a class label that represents its reading. For instance, if the gauge's needle points to '6', we put it in class '6'.

C. Models

As stated earlier, we explored and compared 3 hybrid approaches in this work, namely, OpenCV with Scikit-Image, OpenCV with ResNet18, and OpenCV with a VGG19. PyTorch framework was used to train and evaluate the deep Learning models.

1) OpenCV with Scikit-Image: This approach used OpenCV to load and preprocess images and then applied scikit-image for enhanced filtering and region analysis [8]. Scikit-Image is a Python library built on top of NumPy, SciPy, and matplotlib. We tagged this the "Traditional computer vision technique" as it employed a four-stage approach for needle angle detection and classification. The stages are summarized as follows:

- Canny edge detection with low and high thresholds of 50 and 150 respectively, was used to identify the needle boundaries.
- Probabilistic Hough Line Transform was then used to extract the needle line from the detected edges.
- Geometric computation was used to calculate the needle's angle relative to the gauge center.
- The obtained angles were mapped to discrete classes using predefined angular intervals of 32.7 (360 divided by 11) degrees per class, enabling classification across 11 classes spanning the full 360-degree range. This systematic approach helped in uniform distribution across the entire angular spectrum, with each class representing an equal portion of the gauge's measurement range.

2) OpenCV with ResNet18: This hybrid approach leveraged OpenCV and ResNet18 (implemented in PyTorch). OpenCV was used for image loading and preprocessing, specifically for image resizing and conversion to grayscale, while ResNet18 was used for classification. Our implementation process can be summarized as follows:

- The images were converted to to 224x224 pixels and converted to 3-channel grayscale format to meet the input requirements of pre-trained ResNet18.
- We utilized apre-trained ResNet18 backbone for feature extraction, but still trained it on our dataset. By utilizing a pre-trained model we were able to minimize the training time and benefitted from transfer learning [18]. We froze the ResNet model during the first initial 10 epochs before being fine-tuned in subsequent training phases. In total we trained the model using 100 epochs.
- The classification component of our model consisted of a modified final fully connected layer that mapped the 512-dimensional feature representations to 11 output classes. We applied a dropout rate of 0.3 before the final layer to mitigate overfitting and improve generalization performance.

3) OpenCV with VGG19: Like in the previous model, we also applied OpenCV to preprocess the images and convert them to 224×224×3 RGB images. The converted images were then passed as input to a VGG19 model pre-trained on ImageNet for feature extraction.

- The original classifier structure of 25088→4096→4096→1000 was modified to 25088→1024→512→11 to accommodate the 11-class classification task.
- We applied ReLU activation functions and dropout layers throughout the network. Finally, to prevent overfitting we applied a dropout rate of 0.5 across each of the fully connected layers in the modified classifier.

4) Performance Metrics: To evaluate the performance of the model, the following metrics were used:

- Accuracy - Measures how close the digitized value was to the true pressure value (ground truth).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

[15]

- Precision - Measures the consistency of repeated digital equivalent values of the same pressure value of the analogue gauge.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

[15]

- Recall - This measured the model's ability to detect and return the correct digital equivalent value with respect to the pressure value across the entire gauge range.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

[15]

- F1-Score - This computed the harmonic mean of precision and recall for the models to ensure that all readings are captured and there are no missed readings.

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

[15]

- Cross-Entropy Loss - This measured the average difference between predicted probability distribution and true distinct values (labels) readings of the analogue gauges.

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}) \tag{5}$$

[16]

- Mean Average Precision (mAP) - This was used as the primary evaluation metric to assess the performance of the hybrid approaches across all gauge values. It is defined as:

$$AP_i = \int_0^1 P_i(R)\, dR \tag{6}$$

[19]

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{7}$$

[19]

D. Visualization and Analysis Tools

The evaluation methodology encompasses comprehensive performance assessment through multiple visualization and analytical techniques, including confusion matrices to examine class-wise performance and identify misclassification patterns, ROC curves with corresponding AUC values to evaluate binary classification performance for each individual class, training dynamics plots that display loss curves and accuracy progression throughout the training process.

VI. RESULTS AND ANALYSIS

The performance of the three proposed approaches implemented in this work are presented in this section.

A. Comparative Performance Analysis

We start off by comparing ResNet18 and VGG19 as they are both deep learning models. This comparison is important because it highlights the relative strengths and weaknesses of modern convolutional neural network architectures when applied to the task of analogue gauge digitization. ResNet18, with its residual connections, is designed to address the vanishing gradient problem and typically achieves higher accuracy with fewer parameters, while VGG19, although deeper, is computationally more intensive and less efficient in feature reuse. Evaluating these models side by side provides insight into the suitability of deep learning–based feature extraction approaches for this domain. In contrast, the subsequent comparison with Scikit-Image will be based on overall performance rather than per-class metrics. This is because Scikit-Image employs traditional computer vision techniques that rely on handcrafted features rather than learned representations. Such methods do not produce class-wise probabilities in the same way as deep learning classifiers; instead, their effectiveness is better reflected in aggregate accuracy or error rates across the dataset. As a result, reporting overall performance provides a fairer and more meaningful basis for evaluating the relative strengths of classical image processing versus deep learning–based approaches in this application.

TABLE I
COMPARISON OF RESNET18 AND VGG19

Class	Precision		Recall		F1-Score	
	ResNet18	VGG19	ResNet18	VGG19	ResNet18	VGG19
0	1.00	0.57	0.97	0.97	0.97	0.99
1	0.72	1.00	0.94	0.92	0.97	0.35
2	0.96	0.51	0.77	0.40	0.97	0.97
3	0.86	0.57	1.00	0.69	0.94	0.65
4	0.97	0.68	0.94	0.68	0.96	0.68
5	0.94	0.79	1.00	0.68	0.97	0.73
6	1.00	0.67	0.88	0.82	0.94	0.74
7	0.97	1.00	0.97	0.35	0.97	0.52
8	0.96	0.74	0.77	0.59	0.85	0.66
9	0.92	0.43	0.97	0.18	0.94	0.25
10	0.94	0.94	0.97	0.85	0.96	0.89

Table 1 shows that ResNet18 consistently outperformed the VGG19 model in almost all classes in terms of precision, recall and F1-score metrics. The ResNet18 model achieved a near-perfect performance on classes 0, 3, 5, and 7, with F1-score $\geq$ 0.97, demonstrating its robustness in distinguishing fine-grained variations in pointer positions. In contrast,VGG19 was inconsistent, performing strongly in some classes (like class 10 with F1-score of 0.89), but poorly in others (like class 9 with F1-score of 0.25). These results show a significant performance disparity between ResNet18 and VGG19, indicating that ResNet18 was able to generalise and classify readings with a high precision, recall and F1-score.

Figure 5 and 6 further indicate the performance disparity between ResNet18 and VGG19 models. Figure 6 shows that VGG19 exhibited widespread misclassification patterns with significant off-diagonal confusion, especially between classes 2 - 4 and scattered predictions for classes 0, 3, 4, 9, and 10. This dispersion indicates that VGG19 struggles to learn class-specific representations, leading to inconsistent recall across multiple categories. Conversely, ResNet18, depicted in Figure 5, demonstrated a near perfect classification with a strong diagonal pattern indicating minimal confusion between classes with the majority of predictions concentrated along the correct class axis. Misclassifications were minimal and largely limited to a few neighboring classes, such as 5 being confused with 6 or 8 being confused with 9. Such confusions are justifiable, as even humans might also confuse 5 and 6 or

8 and 9, especially when cursive / calligraphy fonts are used. In summary, ResNet18 is able to generalize effectively while maintaining high precision and recall across most categories.

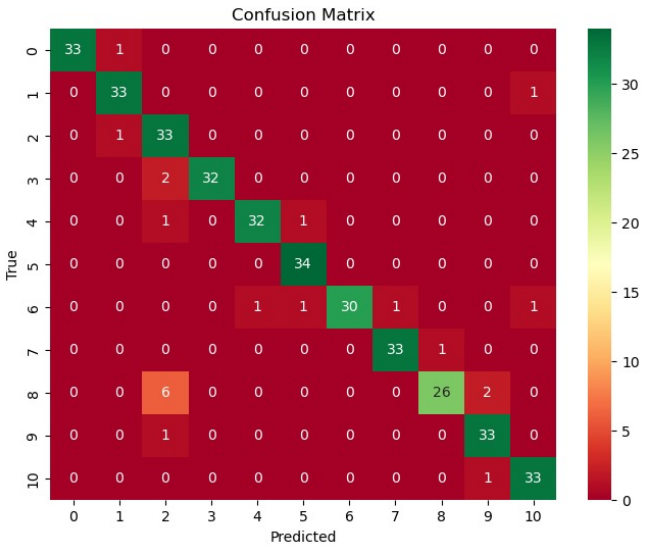


Fig. 5. Confusion matrix for ResNet18.

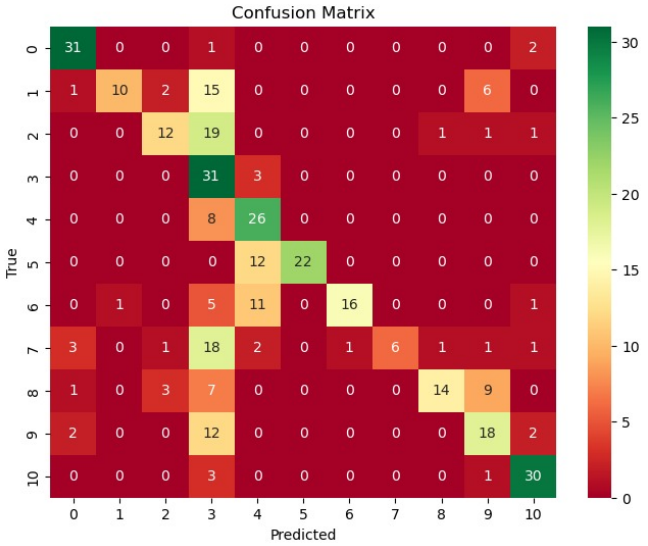


Fig. 6. Confusion matrix for VGG19.

The ROC curves, shown in Figures 7 and 8, further reinforce the differences observed between VGG19 and ResNet18. For VGG19, while the overall area under the curve (AUC) values appear close to 1.0 for all classes, the curve shapes indicate a degree of instability in classification confidence, which aligns with the misclassifications evident in the confusion matrix (Figure 6). In contrast, ResNet18 consistently achieved near-perfect AUC values of 1.0 across all classes, with the curves tightly hugging the top-left corner of the plot. This

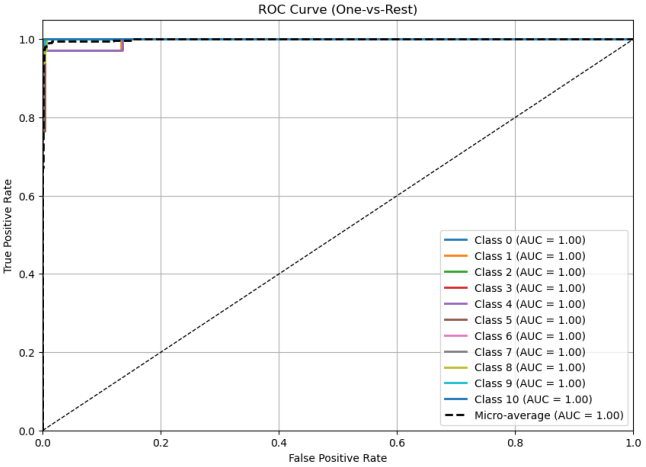


Fig. 7. ROC Curve for ResNet18.

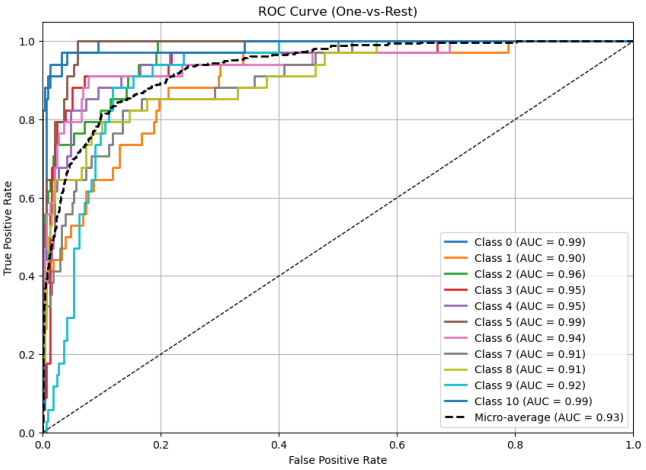


Fig. 8. ROC Curve for ResNet18.

indicates near-perfect separation between true positives and false positives, reflecting both high sensitivity and specificity.

B. Error Analysis

Figures 10 and 11 display the training dynamics between ResNet18 and VGG19, revealing fundamental differences in their learning capabilities and optimization characteristics. ResNet18 converged with its training loss rapidly decreasing from approximately 25 to near-zero within the first 20 epochs. This was followed by a stable low-loss training throughout the remaining epochs. Correspondingly, its validation accuracy quickly reaches and maintains 95-97% with relatively minor fluctuations, indicating robust generalization and effective learning. VGG19, on the other hand, plateaued around 1.4 - 1.6 after initial descent, suggesting convergence to a suboptimal local minimum as its validation accuracy stagnates at only 55-65% with high volatility, indicating poor generalization and potential overfitting despite the higher training loss. While the comparison between ResNet18 and VGG19 highlighted

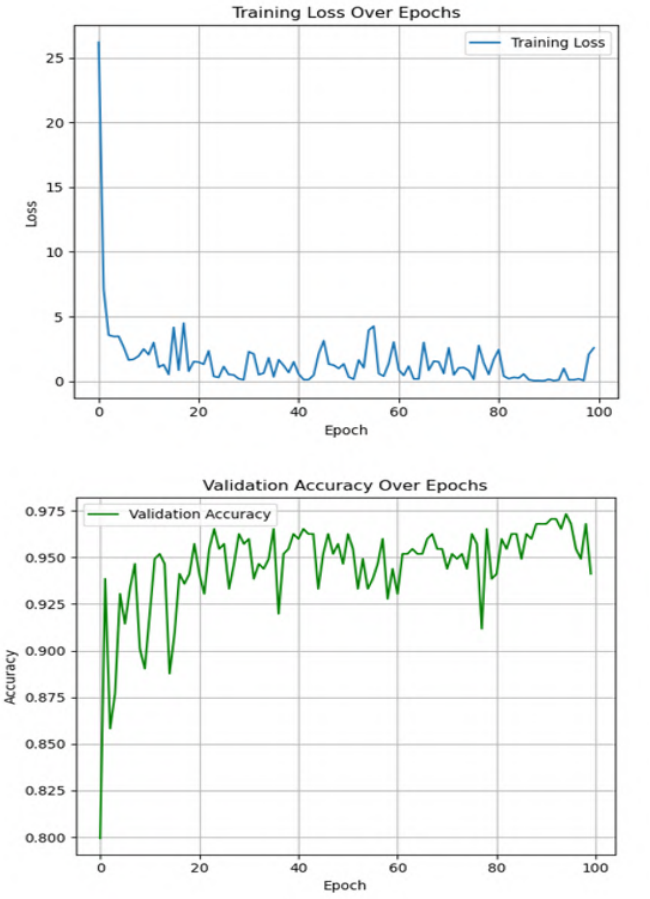


Fig. 9. Training loss and Validation Accuracy over epochs for ResNet18.

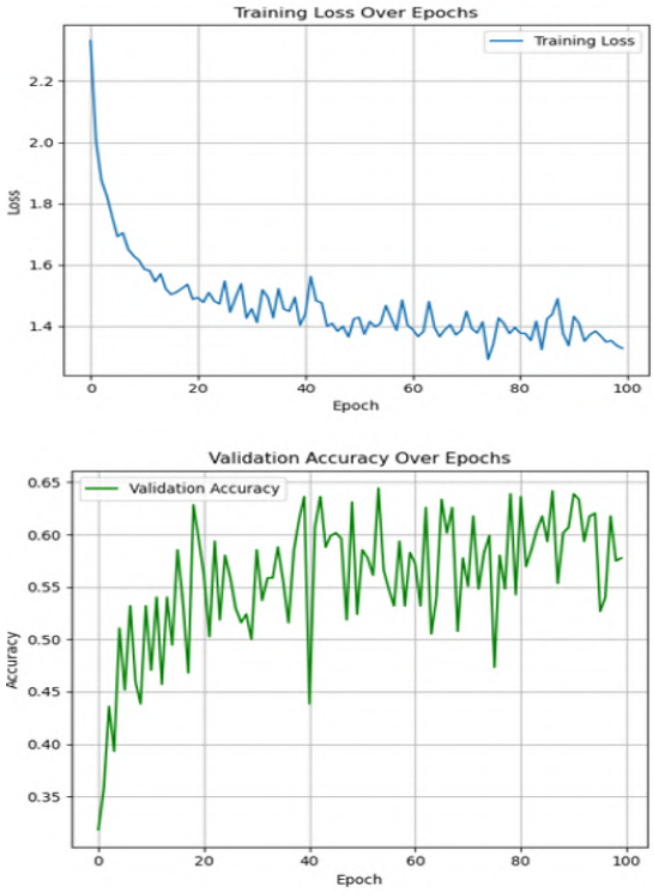


Fig. 10. Training loss and Validation Accuracy over epochs for VGG19.

the superiority of ResNet18 at the class level, it is equally important to contrast deep learning approaches with traditional image processing techniques such as Scikit-Image. Unlike neural networks, which learn representations from data, Scikit-Image relies on handcrafted feature extraction methods and classical algorithms to interpret gauge readings [17]. As a result, its performance is best evaluated in terms of overall accuracy and error rates rather than per-class breakdowns, since it lacks the adaptive capacity to handle nuanced variations across individual classes (which would have been shown with a confusion matrix). This broader evaluation provides valuable insights into whether the added complexity and computational cost of deep learning models are justified by their gains in precision, recall, and robustness when compared to a lightweight traditional approach. The ResNet18 approach demonstrated exceptional performance across all metrics, having achieved near-perfect accuracy (97.3%) and mAP (99.1%). This indicated robust feature learning and generalization capabilities of the model through transfer learning from the pre-trained ImageNet model. Despite having shown perfect precision, the Scikit-Image approach demonstrated significantly lower recall (63.8%), suggesting that it was conservative in making predictions when it classified the classes, it was always correct, but it missed many actual cases, such that the method only successfully identified and classified 63.8% of the actual needle positions presented in the test dataset. This conservative behavior is typical of traditional computer vision methods with strict thresholding [18]. Despite being a deep learning approach, VGG19 performed poorly compared to ResNet18. It achieved a slightly higher accuracy (64.4%) compared to the Scikit-Image approach. However, VGG19 performed poorly in precision, recall, F1-score and mAP compared to the other two approaches (OpenCV with ResNet18 and OpenCV with Scikit-Image).

Table 2 presents a summarised comparison between the 3 approaches. The mAP was used to obtain the results on the table.

TABLE II
COMPARISON OF RESNET18 VS VGG19 VS SCIKIT-IMAGE

Approach	Accuracy	Precision	Recall	F1-score	mAP
OpenCV + ResNet18	97.3%	94.7%	94.1%	94.1%	99.1%
OpenCV + VGG19	64.4%	71.1%	64.4%	63.0%	75.6%
OpenCV + Scikit-Image	63.8%	100%	63.8%	77.9%	100%

C. Computational Performance

We also compared the computational resource and time required by the three models, as summarized in Table 3.

TABLE III
RESOURCE USAGE AND EFFICIENCY COMPARISON

Metric	ResNet18	VGG19	Scikit-Image
Training Time (hours)	5	8	N/A
Inference Time (ms)	23.4 ± 2.1	41.2 ± 3.8	127.5 ± 15.2
Memory Usage (MB)	145	289	23
Model Size (MB)	42.7	532.6	N/A

The table shows that ResNet18 took about 46% less time to train than VGG19. Training times are not available for Scikit-Image because it is not a neural network model, hence does not require training.

When exposed to new images, ResNet18 was almost twice as fast as VGG19 in performing inference (i.e., classifying the image). Scikit-Image, being based on numerical computations, was extremely slow in carrying out this task. It however made up for it with its very small memory footprint, requiring 6x and 12x less memory space than ResNet18 and VGG19 respectively.

VII. DESIGN AND PROTOTYPING

In this section we discuss potential design and implementation of a system that can leverage our proposed models.

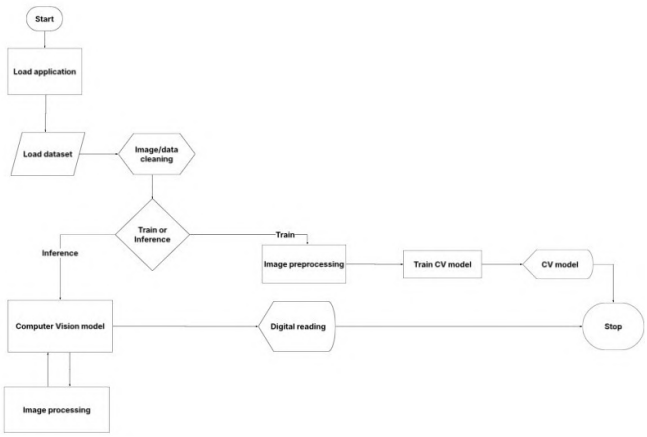


Fig. 11. FlowChart of the system.

A. System Flow

The flowchart in Figure 11 illustrates the process of digitizing analogue gauges using a computer vision model. It begins with loading the application and dataset, followed by image cleaning. Depending on the mode (training or inference), the workflow branches out either into a Training mode - where the cleaned dataset undergoes image preprocessing before training the computer vision (CV) model; Inference mode where preprocessed images are fed directly into the trained CV model, which performs inference and displays the obtained digital readings.The process loops back if needed or stops upon completion.

Note that, in this context, the term "computer vision or CV" refers to any of our proposed models (OpenCV + ResNet18, OpenCV + VGG19, or OpenCV + Scikit-Image).

The workflow supports iterative improvement, such that, if inference results are unsatisfactory, the system can loop back to training with additional data. This pipeline accommodates both traditional CV techniques and deep learning, balancing accuracy and computational efficiency.

Implementation wise, the proposed design can be deployed on a mobile device or single board computer with an attached camera to capture images of the analogue gauges.

B. Interface Design

A medium fidelity prototype was designed using Penport, an internet based prototyping tool [10]. Snapshots of the prototype as shown in Figure 12.

Users need to be authenticated to use the system, as shown in Figure 12 (top left image). On first use, users are encouraged to select their desired approach to get inference under model. This can be done via the Settings page shown in the top right image of Figure 12.

When an image of the analogue gauge has been successfully captured in the system Figure 12 (lower left image), the user can then press the analyze button which will run the selected approach and return the digital equivalent value, as shown in the lower right image in Figure 12.

VIII. CONCLUSION

This study demonstrated the feasibility of digitizing analogue pressure gauges using computer vision. Three hybrid approaches were considered in this work. A traditional CV approach, "'OpenCV with Scikit-image', and two contemporary models based on deep learning, "'OpenCV with ResNet18' and 'OpenCV with VGG19'. For all three approaches, OpenCV was used for image preparation, while the core model was used for inference. We used the three approaches to digitize 10 BAR analogue pressure gauges and compared their performance. The results obtained show that ResNet18 clearly outperformed both VGG19 and the traditional Scikit-Image-based model. The OpenCV+ResNet18 model achieved an average precision of 97.3%and an average F1-score of 94.1% across all classes, compared to OpenCV+VGG19's 64.4% 63% and OpenCV+Scikit-Image's 63.8% and 77.9%. Confusion matrix and ROC analyses also buttressed ResNet18's robustness, with near-perfect class separation. Though Scikit-Image offered high precision, its limited recall and lower overall performance highlight the insufficiency of classical image processing for robust gauge digitization. These findings establish deep learning, and specifically ResNet18, as a practical and effective solution for industrial monitoring, enabling accurate, automated gauge readings, minimizing human error, and supporting Industry 5.0 objectives through predictive maintenance and data-driven decision-making.

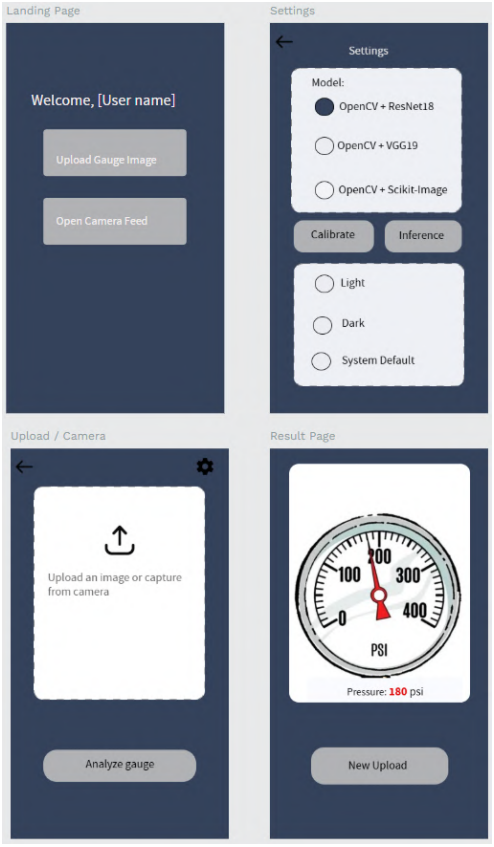


Fig. 12. Medium-fidelity Prototype design .

While this study focused on 10-BAR pressure gauges, the methodology shows strong potential for broader applicability. With targeted fine-tuning, the pretrained ResNet18 and VGG19 models could be adapted to other analogue instruments such as temperature, voltage, or flow meters, opening pathways for unified, automated instrumentation monitoring across multiple domains. Overall, the results establish a solid foundation for future research on transferable gauge-reading systems and mark an important step toward intelligent, data-driven industrial environments.

REFERENCES

[1] R. Archana and P. E. Jeevaraj, "Deep learning models for digital image processing: a review," *Artificial Intelligence Review*, vol. 57, no. 1, p. 11, 2024.

[2] Y. Liang, Y. Liao, S. Li, W. Wu, T. Qiu, and W. Zhang, "Research on water meter reading recognition based on deep learning," *Scientific Reports*, vol. 12, no. 1, p. 12861, 2022.

[3] H. Ninama, J. Raikwal, A. Ravuri, D. Sukheja, S. K. Bhoi, N. Z. Jhanjhi, *et al.*, "Computer vision and deep transfer learning for automatic gauge reading detection," *Scientific Reports*, vol. 14, no. 1, p. 23019, 2024.

[4] S. Chavan, X. Yu, and J. Saniie, "High precision analog gauge reader using optical flow and computer vision," in *Proc. IEEE Int. Conf. Electro Information Technology (eIT)*, pp. 171–175, May 2022.

[5] J. Peixoto, J. Sousa, R. Carvalho, G. Santos, J. Mendes, R. Cardoso, and A. Reis, "Development of an analog gauge reading solution based on computer vision and deep learning for an IoT application," *Telecom*, vol. 3, no. 4, pp. 564–580, Oct. 2022.

[6] B. Howells, J. Charles, and R. Cipolla, "Real-time analogue gauge transcription on mobile phone," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 2369–2377, 2021.

[7] C. Kanan and G. W. Cottrell, "Color-to-grayscale: does the method matter in image recognition?," *PLoS One*, vol. 7, no. 1, p. e29740, 2012.

[8] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, *et al.*, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, 2014.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[10] Penpot, "DAGCV, Penpot Design Workspace," 2024. [Online]. Available: https://design.penpot.app/#/workspace?team-id=921945d0-8932-8162-8006-2492c33772c7&file-id=95aea877-d515-80ac-8006-24932424a647&page-id=95aea877-d515-80ac-8006-1f7b16b956f8

[11] X. Yang, "Machine Learning in Practice: Using Artificial Intelligence to Read Analog Gauges," Object Computing, St. Louis, MO, USA, Jun. 2019. [Online]. Available: https://objectcomputing.com/resources/publications/sett/june-2019-using-machine-learning-to-read-analog-gauges

[12] Third Eye Data, "The Hidden Cost of Manual Gauge Readings: Explaining the AI Solutions for Manufacturing Industry," May 6, 2024. [Online]. Available: https://thirdeyedata.ai/the-hidden-cost-of-manual-gauge-readings-and-the-ai-solutions/

[13] Beamex, "Manual Data Entry Errors," 2021. [Online]. Available: https://blog.beamex.com/manual-data-entry-errors

[14] Ocrolus, "8 Human Error Examples and Statistics," Dec. 17, 2024. [Online]. Available: https://www.ocrolus.com/blog/human-error-8-eye-popping-sets-of-stats-and-examples/

[15] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[16] C. Mao, M. Mohri, and Y. Zhong, "Cross-entropy loss functions: Theoretical analysis and applications," in *Proc. Mach. Learn. Res.*, vol. 206, pp. 6069–6108, 2023.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770–778, 2016.

[18] L. Ntantiso, A. Bagula, O. Ajayi, and F. Kahenga-Ngongo, "A Review of Federated Learning: Algorithms, Frameworks and Applications," in R. A. Saeed, A. D. Bakari, and Y. H. Sheikh, Eds., *Towards new e-Infrastructure and e-Services for Developing Countries (AFRICOMM 2022)*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 499. Cham: Springer, 2023, pp. 1–20. doi: 10.1007/978-3-031-34896-9_20

[19] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge 2012 (VOC2012) results," 2012.

[20] A. R. Rizaman, H. Selamat, and N. Khamis, "Digitization of analogue meter reading using convolution neural network," *Mekatronika: Journal of Intelligent Manufacturing and Mechatronics*, vol. 3, no. 1, pp. 28–34, 2021.

[21] J. Leon-Alcazar, Y. Alnumay, C. Zheng, H. Trigui, S. Patel, and B. Ghanem, "Learning to read analog gauges from synthetic data," in *Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV)*, pp. 8616–8625, 2024.

A Chronological Analysis of the Evolution of SmartNICs

1st Olasupo Ajayi
CAESAR Laboratory,
Electrical & Computer Engineering Department,
Queen's University, Kingston,
ON, K7L3N6, Canada
0000-0001-6583-3749

2nd Ryan Grant
CAESAR Laboratory,
Electrical & Computer Engineering Department,
Queen's University, Kingston,
ON, K7L3N6, Canada
0000-0002-0163-3892

***Abstract*—Network Interface Cards (NICs) are one of the key enablers of the modern Internet. They serve as gateways for connecting computing devices to networks for the exchange of data with other devices. Recently, the pervasive nature of Internet-enabled devices coupled with the growing demands for faster network access have necessitated the enhancement of NICs to Smart NICs (SNICs), capable of processing enormous volumes of data at near real-time speed. However, despite their popularity, the exact use and applicability of SNICs remains an ongoing debate. These debates are exacerbated by the incorporation of accelerators into SNIC, allowing them to relieve their host's CPUs of various tasks. In this work, we carry out a chronological analysis of SNICs, using 370 articles published in the past 15 years, from 2010 to 2024, to gain some insight into SNICs; and shed some light on their evolution, manufacturers, use cases, and application domains.**

***Index Terms*—Data Analysis, DPU, FPGA, HPC, Network Interface Cards, NIC, SmartNIC**

I. Introduction

Smart Network Interface Cards (SNICs) are a type of network interface card (NIC) that are equipped with a system-on-chip (SoC) capable of running full-fledged operating systems and can be programmed to handle various tasks, thus offloading such tasks from the host's Central Processing Unit (CPU) [1], [2].

Recently, several terminologies have emerged around the concept of SNICs, including Network Accelerators, Data Processing Unit (DPU), and FPGA-NIC (Field Programmable Gateway Array - NIC). There has also been a growing debate about the exact purpose and use cases of these devices. In many contexts, especially in High Performance Computing (HPC), SNICs are considered accelerators that are used to offload various tasks off the main CPU. The concepts of network and workload offloading are well discussed in literature hence, not repeated here. Rather, in this work, we carry out a chronological analysis of the concept of SNICs, exploring scholarly publications spanning a 15-year period from 2010 to 2024. The purpose and main contributions of this work are as follows:

- To gain insights into how SNICs are being used in different scenarios, by exploring applications and use cases reported in literature.
- Identify the main research areas where SNICs have been and are being applied.
- Curate a dataset of published works on SNICs spanning a 15-year period.
- Identify the main manufacturers of SNICs and how their products have evolved over the past decade and a half.

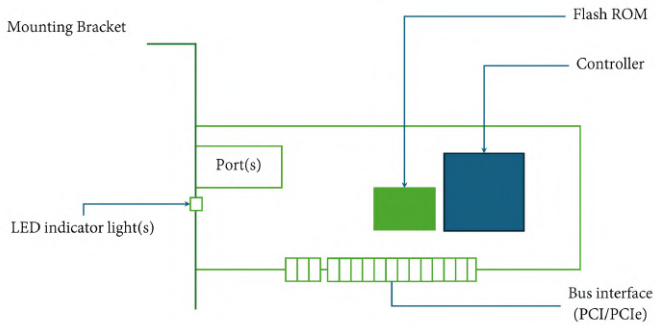
II. Background

Although the Transmission Control Protocol/Internet Protocol (TCP/IP) and Open Systems Interconnection (OSI) stacks have received a lot of attention, NICs are undoubtedly the unsung heroes of modern networks. A NIC (or network adapter) is a hardware component that connects a computing device to a network and allows the device to receive or send data to other devices on the network. NICs operate in both the first and second layers (physical and data link) of the OSI stack or layer 1 (link layer) of the TCP/IP protocol stack, implementing the IEEE 802.3 (wired Ethernet) and IEEE 802.11 (wireless) communication standards.

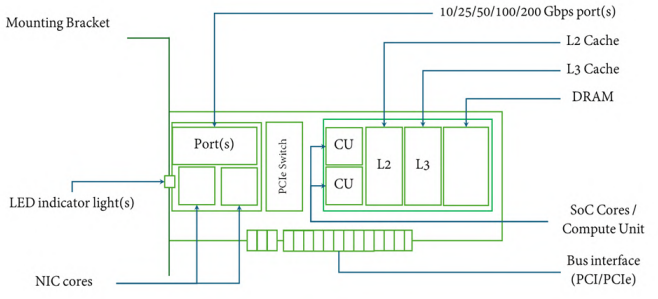
NICs are responsible for host addressing within a network (via Media Access Control or MAC address), data transmission (between its host device and the network), transmission control (including congestion avoidance and prevention of packet collisions), and error detection and correction (via checksums and redundancy checks). Fig. 1 shows a graphical comparison of the basic layouts of a classic NIC and a smart NIC (SNIC).

Computer networks have been growing at an exceptional rate in recent years. This is mainly due to the increase in global internet usage, which, as at the time of writing, stands at approximately 66% of the world's population [3], and the ubiquity of smart phones and connected devices. These factors have led to the evolution of NICs from mere network connection points to more advanced devices fitted with specialized circuitry capable of handling higher data transmission rates, faster processing, and more functionalities, called Smart NICs (SNICs).

From a functionality standpoint, SNICs can be classified into three categories - Application Specific Integrated Circuit (ASIC) based, Field Programmable Gateway Array (FPGA) based, and those that incorporate SoCs, called Data Processing



(a) Generic Network Interface Card Layout



(b) Generic SNIC

Fig. 1: Network Interface Cards

Units (DPUs). ASIC-based SNICs, as the name suggests, are designed for a specific purpose, such as offloading network tasks off the CPU. These NICs are cheaper but not flexible, hence, cannot easily adapt to network needs. FPGA-based SNICs, on the other hand, are flexible and can be programmed for multiple network tasks. DPUs (SoC-based SNICs) have built-in processing units (ARM processors), memory, and host interface, in addition to ASIC or FPGA processing elements. They can be considered synonymous to edge or single-board computers and are highly programmable using C/C++ and/or Programming Protocol-independent Packet Processors (P4) programming languages. DPUs often have multiple accelerators and can be programmed to offload several tasks from their host's CPU [1], [2], [4].

III. Related Works

A comprehensive review of SNICs was carried out in [7]. The work defined "SmartNICs" as programmable NICs, then compared various SNICs based on their design, hardware architecture, and type (ASIC, FPGA, SoC, Hybrid). A survey of the programmability of SmartNICs, using P4, C/C++, and Nvidia's DOCA was also done by the authors. Finally, the work considered some common SmartNICs, specifically Nvidia's Bluefield and Netronome's NFP series, reviewed research publications where they were used, and discussed some of the open challenges and limitations of SmartNICs in general.

In [9], a survey of SNICs was performed from both industrial and research perspectives. The authors reviewed several articles in an effort to propose a universal definition of the term "SmartNIC". This was then followed by an exploration of the various architectures, i.e., ASIC-based, FPGA-based, and SoC-based SNICs. Finally, the work compared two products, Intel's Programmable Acceleration Card (PAC) N3000 and Nvidia's Bluefield-2, and their performance in mitigating Distributed Denial of Service (DDoS) attacks.

In a similar work, [5] surveyed SNICs holistically. The authors began by comparing classic NICs with SmartNICs and then showed the evolution from the former to the latter. The work grouped SNICs into five categories, namely, ASIC-based, FPGA-based, ASIC+FPGA hybrid, ASIC+SoC and FPGA+SoC. Using a taxonomy as a guide, the authors compared various SNICs based on their programmability, CPU architecture, and traffic switching abilities. Task offloading using SNICs was also discussed and grouped into security, network, storage, and computing tasks. The paper was concluded with a discussion on the open challenges of SNICs.

An evolutionary trend analysis of SNICs was carried out in [8]. SNICs from major manufacturers were reviewed, as well as their applications as workload accelerators in various domains, including telecommunications, storage, artificial intelligence, and edge processing. Similarly, [6] also explored the potential of using SNICs as general-purpose accelerators in HPC. They explored the offloading mechanisms of Nvidia's Bluefield-2 DPU using Pennant and BigSort benchmarking applications.

This work, unlike other related works, does not focus on the architectures, taxonomy, or comparison of offloading capabilities of SNICs in HPC; instead, we carry out a systematic chronological analysis of SNICs, using published articles, to provide insights into their evolution and real-world applications over a 15-year period. To the best of our knowledge, this work is the first of its kind with respect to SNICs.

IV. Methodology

Being a systematic review, this work followed the PRISMA guidelines [10], with the following steps taken to carry out the analysis.

- **Step 1 – Data collection:** data on SNICs related publications were curated from IEEE Xplore repository. "DPU", "SmartNIC", and "FPGA-NIC" were used as keywords, with the search limited to articles published in the 15 years between 2010 and 2024.
- **Step 2 – Filtration / screening:** Step 1 resulted in about 490 articles, which were then passed through two filtration phases. In the first filtration phase, we read the abstract and index or key words of each article to filter out irrelevant articles. 26 articles were excluded through this process. In the second phase, we read the entire content of each of the remaining articles. During this phase, we were able to exclude articles relating to smart switches and other solutions that were also termed data processing units (DPU) but referred to systems designed for other forms of end-to-end data processing. Finally, we were left with 370 articles.

- **Step 3 – Pre-processing:** We pre-processed the metadata of each article (in bib and JSON formats) and created a Comma Separated Value (CSV) file using JabRef [11]. The initial CSV file had 30 columns (fields), from which we extracted six (6) relevant fields, namely "Author", "Title", "Year", "Abstract", "Publisher", and "NIC Type". NIC type refers to the type of SNIC used in the article, i.e., "DPU", "FPGA-NIC" or "SmartNIC".
- **Step 4 – Dataset preparation:** We read each article to extract information on the type, model, and manufacturer of the SNIC(s) used by the researcher(s) / author(s). We also extracted information about the application domain (e.g., Medicine, AI, Databases, etc.), and specific focus of the research work (e.g., Energy conservation, Edge Processing, Performance evaluation and benchmarking, etc.). Finally, we included four (4) additional fields in our dataset, namely "Manufacturer", "Device", "Research Focus", and "Application Domain". In total, we had 10 fields and 370 rows in our dataset. The dataset curated from this work is available at [12].
- **Step 5 – Data analysis:** In this step, we performed a chronological analysis of the data using a combination of Python, Microsoft Excel and PowerBI. For each analysis, the dataset (of articles) was grouped into 3 bins, each spanning a 5-year interval - "2010–2014", "2015–2019" and "2020–2024".

V. Research Findings

As stated above, the primary objective of this work was to perform a chronological analysis of the evolution of SNICs over a 15-year period, divided into 5-year bins. These bins would be used as guardrails for presenting the finds.

In the results presented in this section, the terms "DPU", "SmartNIC" and "FPGA-NIC" refer to the terms explicitly used by the authors of each publication. To avoid confusion between "SmartNIC" and "SNIC", in this work, "SNIC" refers to a general term used to categorize all forms of "smart" NICs, i.e., ASIC / FPGA / SoC based NICs, as described in section 2.

A. Device Type

The distribution of devices by type is shown in Fig. 2. Although the figure shows that "SmartNICs" are the most common, this was not the case until 2020–2024. Fig. 3 reveals that "SmartNICs" did not exist between 2010-2014, while "DPUs" was only reported in 3 articles [13]–[15]. Also, before 2020, "FPGA-NICs" were the most common type of SNIC. The exponential rise of "DPUs" and "SmartNICs" post 2020 is likely driven by the acquisition of Mellanox by Nvidia in 2020 and the emergence of its Bluefield brand of NICs.

B. Device Manufacturers

During the 15-year span, Fig. 4 shows that AMD was the dominant manufacturer. Even with the arrival of Nvidia in the 2020-2024 time frame, AMD still maintained almost 40% of the market share. Nvidia, Netronome, and Intel were,

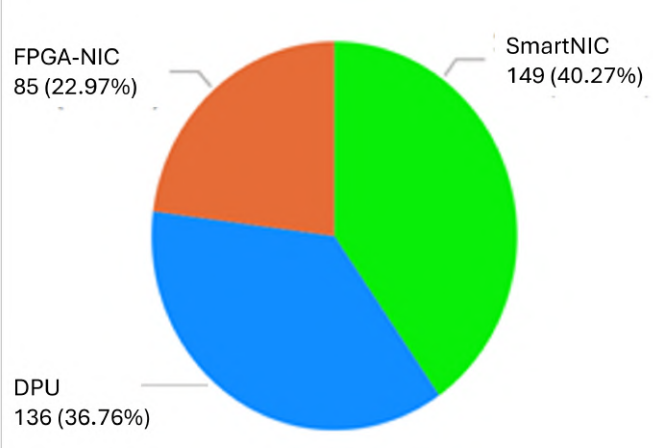


Fig. 2: Device Types

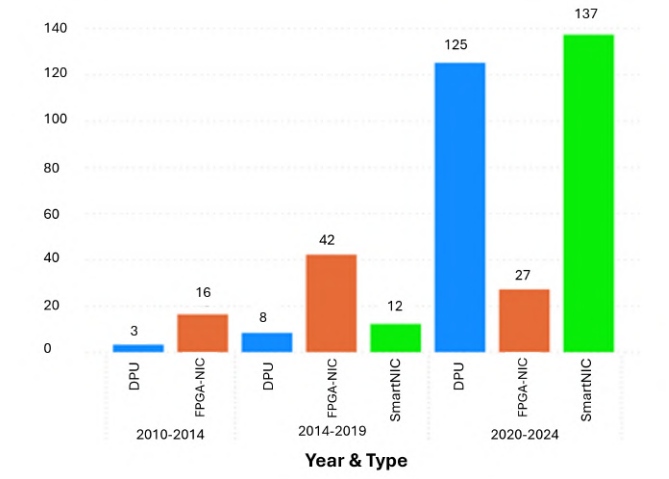


Fig. 3: Device Type distribution over the past 15 years

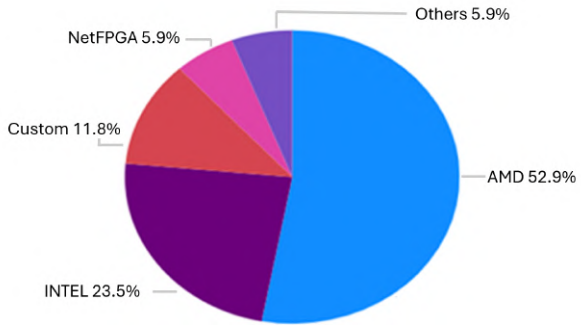
respectively, the second, third, and fourth largest SNICs manufacturers in the past 15 years.

C. Device Models

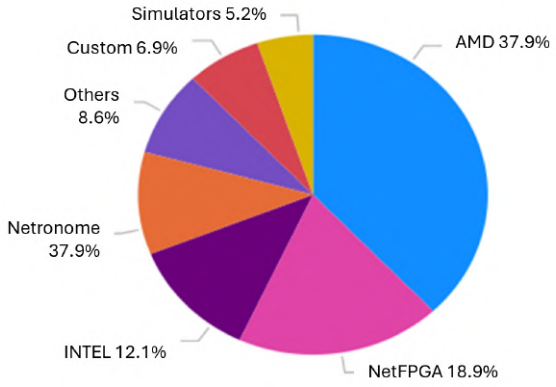
Devices of different types and models were used in the 370 articles reviewed. In some cases, the authors provided the exact device and model used, e.g., "Netronome Agilio NFP-4000 CX Dual-Port 10 Gigabit", while in other cases only the device family was mentioned, e.g., "Xilinx Virtex 7" or "NFP-4000". For this reason, our report has a mix of specific device model and family.

Please note that the graphs shown in Fig. 5 are truncated and only show the top-N devices for each time frame. This was done for space-saving purposes. Table I provides information on the number of unique device models per time frame, using a truncation thresholds of at least 2, $(N >= 2)$.

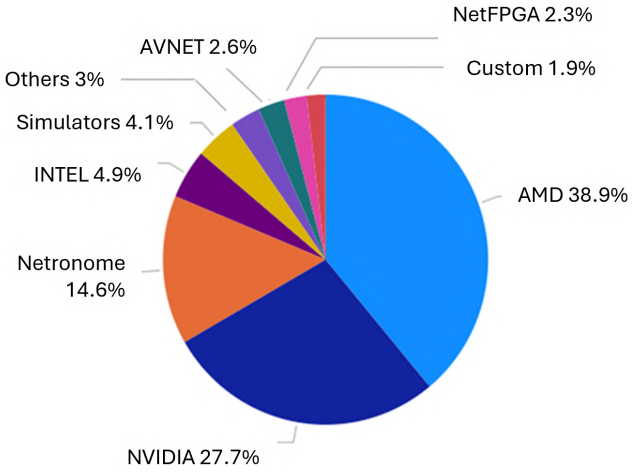
Between 2010 and 2014, Fig. 5a shows that three device models (AMD Xilinx Virtex-5, Intel Altera Stratix, and custom) were prominent. Custom refer to devices that were built



(a) Major Manufacturers (2010–2014)

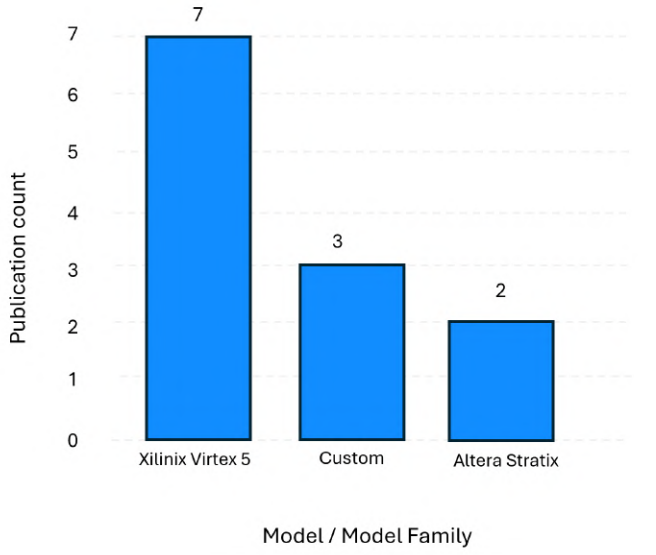


(b) Major Manufacturers (2015–2019)

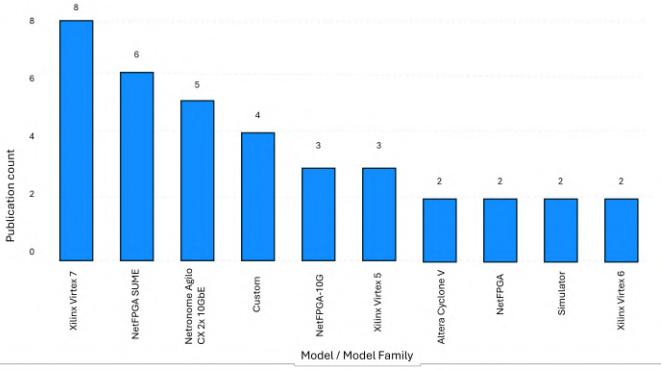


(c) Major Manufacturers (2020–2024)

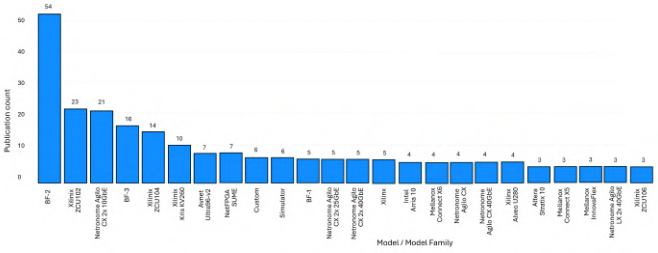
Fig. 4: Device Manufacturers across the 15-year span



(a) Most Common Device Models (2010–2014)



(b) Most Common Device Models (2015–2019)



(c) Most Common Device Models (2020–2024)

Fig. 5: Top-N SNIC Devices used between 2010-2024

by the authors, e.g., NanoBFP [35]. Other devices used in this time frame include Intel's Altera Cyclone III, Xilinx Virtex-6, NetFPGA 10G, etc. Since each of these devices were reported in less than 2 publications, they were not shown in the graph.

Fig. 5b shows the top 10 devices used in the 2015-2019 time frame. AMD's Xilinx Virtex-7 (8) was the most common, followed by NetFPGA SUME, used in 6 research works, and Netronome's Agilio CX dual port 10GbE SmartNIC (5). Although "NetFPGA SUME" and "NetFPGA 10G" use Xilinx Virtex-7 and Virtex-5 chips, respectively, we do not consider them as part of the AMD Xilinx family. Finally, "Simulation" refers to research work in which software simulation tools, such as Vivado [36] and sPIN [37], were used instead of physical SNICs.

The 2020-2024 time frame was dominated by "DPU", which cumulatively represented more than 50% of all devices.

TABLE I: Summary of device count per time frame

Time frame	Unique Device Models	Top N (Truncation point)
2010 - 2014	11	N: $count \leq 2$
2015 - 2019	32	N: $count \leq 2$
2020 - 2024	94	N: $count \leq 3$

These include Nvidia's Bluefield-2 (54) and Bluefield-3 (16), and AMD's ZCU 102 (23) and ZCU 104 (14) DPUs. The Netronome Agilio CX 2x10GbE SmartNIC was also common, as it was used in about 21 publications. The graph also shows that 4 articles used "Netronome Agilio CX", unfortunately, the authors did not specify the exact device model within the Agilio CX family of SmartNICs they used. Although only 24 devices are shown in Fig. 5c, a total of 94 unique devices were used in this period. The remaining 70 are not shown because they were used in less than 4 research work each ($N <= 3$ in Table I).

D. Research Focus

Fig. 6a shows that, between 2010 and 2014, researchers mostly focused on the architecture and design (Archi/Design) of FPGAs to improve network functionality, specifically reducing network latency. The few offloading tasks reported in this time frame were applied to Message Passing Interface (MPI) in HPC [16], memory (direct memory access), database processing and artificial intelligence in medical applications [13].

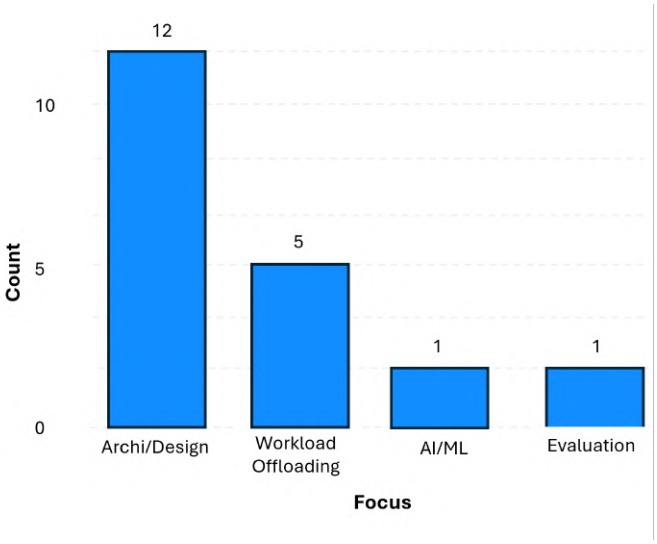
In addition to the four research areas in 2010-2014, programming of network devices using the P4 language [17], [18] and Edge processing [19], [20] were also investigated between 2015 and 2019, as shown in Fig. 6b. In the same period, the offloading of workloads onto SNICs became mainstream, with most offloading tasks geared towards security [21] and databases [22]. Similarly, the use of these devices to design and run Artificial Intelligence & Machine Learning tasks (AI/ML) [23], [24] also became common.

By 2020-2024, SNICs ("SmartNIC", "DPUs", and "FPGA-NIC") had become well established as tools for offloading various workloads and running AI/ML tasks. Fig. 6c shows that these two research areas represented 70% (40% and 30% respectively) of all research work related to SNICs in that time frame. In the same period, the application of SNICs in energy and power conservation [25], [26] as well as storage [27], [28] also become popular.

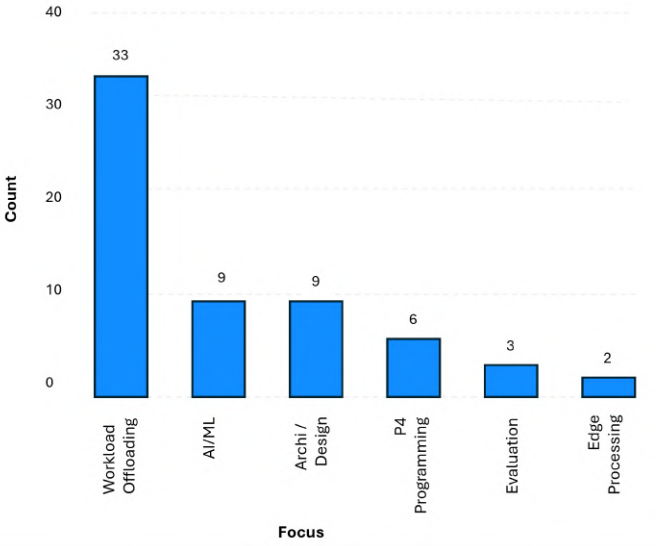
E. Application Domains / Use Cases

Table II provides a concise distribution of the various use cases of SNICs ("DPUs", "FPGA-NIC", and "SmartNIC") between 2010 and 2024.

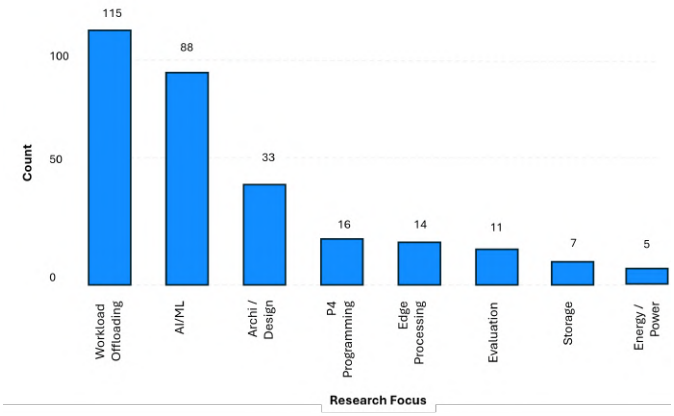
In perusing the table, it immediately becomes clear that SNICs are pervasive and have been applied in numerous domains, beyond just networking. This is due to the offloading capabilities of SNICs, specifically the "DPU" and "SmartNIC". During the 15-year span, the highest number of non-network-specific use cases occurred in the most recent years (2020–2024).



(a) Major Research Focus (2010 – 2014)



(b) Major Research Focus (2015 – 2019)



(c) Major Research Focus (2020 – 2024)

Fig. 6: Distribution of Research Focus across the 15-year span

TABLE II: Applications of SNICs across various Domains from 2010 to 2024

Application domain / Use cases	Years								
	2010-2014		2015-2019			2020-2024			
	DPU	FPGA-NIC	DPU	FPGA-NIC	SmartNIC	DPU	FPGA-NIC	SmartNIC	Total
5G / 6G / Wireless Networks				2		3		10	15
AI/ML Specific Tasks				3		16	3	6	28
Autonomous Vehicles & Automobile			1	2		10		2	15
Blockchain				2				2	4
Cache / Memory		5	2	5		7	6	8	33
Cloud Computing						1	1	8	10
Data / Databases		3	1	8		7	3	13	35
Edge Processing						10		1	11
Energy / Power			1			3			4
Generic Tasks				2		2	2	3	9
Graph Processing						2			2
Human Activity Detection						2			2
Medicine	1	1	2			6		2	12
Molecular Dynamics						1		1	2
MPI / HPC		1				3		4	8
Network Processes		2		7	1	5	3	28	46
Performance Benchmarking / Improvement	1	1		2	2	24	2	8	40
Resource Management						2	1	1	4
Robotics								2	2
SDN/NFV				2	5	1	1	11	20
Security	1	3	1	7	4	18	5	27	66
Task Scheduling						2			2
Total	3	16	8	42	12	125	27	137	370

"Security" had the highest number of applications (66 or 18% of all articles reviewed). This is unsurprising, as these use cases were aimed at securing networks. "Network Processes" had the second highest (46) use case. Again, this comes as no surprise, as SNICs are primarily network devices, and network processes include tasks such as network compression, filtration and firewalls. Interestingly, many researchers focused on "Performance Benchmarking / Improvement" (40), that is, analyzing and benchmarking the performance of SNICs. This implies that about 11% of all published work focused on the design / architecture of various forms of SNICs and benchmarking these variants against the state of the art.

Despite the popularity of AI and ML, only 7.5% of the publications reviewed discussed the application of SNICs to AI/ML specific tasks. To avoid confusion, in this context, "AI/ML Specific Tasks" refer to tasks that are solely associated with AI/ML. These include training [29], [30], quantization [31], [32], and pruning [33], [34] of AI/ML models on SNICs. The limited applications of SNICs in this domain is understandable, as SNICs are resource constrained with modest storage and processing capabilities, thus, are generally ill-suited for most AI/ML specific tasks, which usually require enormous computing resources.

VI. Insight & Discussions

Between 2010 and 2024, the following insights can be drawn about SNICs:

- FPGA-based and SoC-based SNICs were the most common type of SNICs due to their programmability and versatility.
- FPGAs are a major technological pedestal upon which modern day SNICs are built. This is because most "DPUs" and "SmartNICs", except Nvidia Bluefield 2 and 3, are hybridized forms of FPGA + SoC NICs.
- Regarding the terminology debate, the choice between "SmartNIC" versus "DPU" is mainly the manufacturer's preference. Considering the top 4 brands, Netronome uses the term "SmartNIC", while AMD uses both terms - "SmartNIC" for its Alveo series and "DPU" for its Pensando series. Nvidia uses the term "SmartNIC" for its Connect-X series and "DPU" and/or "SuperNIC" for its Bluefield series. Intel uses the term "PAC" for its general or application-specific workload accelerators and Infrastructure Processing Unit (IPU) for its NICs developed for offloading infrastructure management tasks, such as networking, storage, and security.
- Many of the smaller SNIC manufacturers utilize AMD's FPGAs as their core. For example, NetFPGA's SUME and NetFPGA-10G NICs use the Xilinx Virtex-7 and Virtex-5 FPGAs respectively. Similarly, Napatech NT200AS and NT40A01 NIC, respectively, use Xilinx XCVU5P and Xilinx KU11P FPGA, while Alpha-Delta ADM-PCIE-9H3 uses Xilinx XCVU33P. Netcope's NFB-200G2QL uses Xilinx XCVU7P, while Hitech Global's HTG-V5TXT is built on AMD's Xilinx Virtex-5 FPGA.
- The rise of "DPU" and "SmartNIC" has expanded the use of SNICs from pure networking tools to generalized devices (accelerators) capable of offloading different tasks from the host CPU. SNICs are now ubiquitous and are increasingly being applied in other seemingly unconventional application domains, such as autonomous vehicles, robotics, and medicine.

It is important to note that the manufacturers and devices mentioned in this work do not depict the entire landscape of

SNICs, as this work is based on research articles published only in IEEE Xplore repository. There are numerous other manufacturers, such as Marvell - LiquidIO SmartNICs, Broadcom - Stingray, Cisco – Nexus, etc., that were not presented in this work. In addition, we have also identified about 1,500 related publications in ACM repository, which when analyzed might buttress or skew the results presented in this work.

VII. Conclusion

In this paper, we performed a chronological analysis of the evolution of Smart Network Interface Cards (SNICs) over a 15-year period, from 2010 to 2024. 370 articles on SNICs published in IEEE Xplore repository were reviewed to gain insight into the evolution of SNICs and their applications in various domains. We identified the four main SNIC manufacturers, which are AMD, Nvidia, Intel, and Netronome, and their most prominent device models. We also discovered an exponential increase in the popularity of SNICs, specifically DPUs, between 2020–2024, and their general acceptance as accelerators to offload diverse workloads from the CPUs of their hosts.

Although this work unearthed several useful insights about SNICs, it only considered articles published in IEEE Xplore. The exploration of other repositories, specifically ACM, could be a potential avenue to extend this work. Additionally, the use of SNICs in space and orbital satellites and in smart network switches was not considered in this work. These are opportunities for potential future work.

References

[1] Miano, S., Doriguzzi-Corin, R., Risso, F., Siracusa, D. and Sommese, R. (2019). Introducing SmartNICs in Server-Based Data Plane Processing: The DDoS Mitigation Use Case. IEEE Access, 7, pp. 107161-107170, 2019, doi:10.1109/ACCESS.2019.2933491.

[2] Bhalgat, A. (2021, Sept. 14). Choosing the Best SmartNIC. Technical Blog. https://developer.nvidia.com/blog/choosing-the-best-dpu-based-smartnic/

[3] Kemp, S. (2024, Jan. 31). Internet Use in 2024. DATAREPORTAL https://datareportal.com/reports/digital-2024-deep-dive-the-state-of-internet-adoption

[4] Chi, Y., Cui, W., Qiao, L., Long, X. and Wang, L. (2019). Hardware Architecture Design of the Deep-learning-based Machine Vision Chip. IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 2019, pp. 1110-1113, doi:10.1109/ITNEC.2019.8729501.

[5] Kfoury, E., Choueiri, S., Mazloum, A., AlSabeh, A., Gomez, J., and Crichigno, J. (2024). A Comprehensive Survey on SmartNICs: Architectures, Development Models, Applications, and Research Directions. IEEE Access, vol. 12, pp. 107297-107336, 2024, doi:10.1109/ACCESS.2024.3437203

[6] Williams, B., Chen, Y., Poole, W. and Poole, S. (2023). Exploring Challenges Associated with Employing SmartNICs as General-Purpose HPC Accelerators. 2023 IEEE High Performance Extreme Computing Conference (HPEC), Boston, MA, USA, 2023, pp. 1-7, doi:10.1109/HPEC58863.2023.10363618.

[7] Luizelli, M., Vogt, F., De Matos, G., Cordeiro, W., et al. SmartNICs: The Next Leap in Networking. https://smartness2030.tech/wp-content/uploads/2024/05/Ch2.pdf

[8] Boddupalli, R. and Sable, N. (2023). Smart NICs – Evolution and Future Trends. Intel. https://www.intel.in/content/dam/www/central-libraries/xa/en/documents/2023-12/soc-track-smart-nics-1-.pdf

[9] Döring, T., Stubbe, H., and Holzinger, K. (2021). SmartNICs: Current trends in research and industry. Network, 19. doi:10.2313/Net-2021-05-1_05

[10] Page, M., McKenzie, J., Bossuyt, P., Boutron, I., et al. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. bmj, 372. https://doi.org/10.1136/bmj.n71

[11] Kopp, O., Snethlage, C. and Schwentker, C. (2023). JabRef: BibTeX-based literature management software. TUGboat, 44(138), 441–447. https://doi.org/10.47397/tb/44-3/tb138kopp-jabref

[12] Ajayi, O. (2025). 15 Year Evolution of Smart NICs. IEEE Dataport. https://dx.doi.org/10.21227/nwxe-ct19

[13] Chen, Y., Wang, T. and Hu, Z. (2013, October). Implementation of an embedded dual-core processor for portable medical electronics applications. In 2013 IEEE 10th Int. Conf. on ASIC (pp. 1-4). IEEE.

[14] Rao, X., Wang, T. and Wang, Y. (2013, October). A low-power and high-efficiency cache design for embedded bus-based symmetric multiprocessors. In 2013 IEEE 10th Int. Conf. on ASIC (pp. 1-4). IEEE.

[15] Naghash, F. and Eshghi, M. (2014, May). Design an ASIP for Rabbit and HC-128 algorithms. In 2014 22nd Iranian Conf. on Electrical Engineering (ICEE) (pp. 1174-1179). IEEE.

[16] Tanabe, N. and Nakajo, H. (2010, April). Acceleration for MPI derived datatypes using an enhancer of memory and network. In 2010 IEEE Int. Symp. on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW) (pp. 1-6). IEEE.

[17] Mohammadkhan, A., Panda, S., Kulkarni, S., Ramakrishnan, K. and Bhuyan, L. (2019, November). P4NFV: P4 enabled NFV systems with SmartNICs. In 2019 IEEE conf. on network function virtualization and software defined networks (NFV-SDN) (pp. 1-7). IEEE.

[18] Eran, H., Zeno, L., István, Z. and Silberstein, M. (2019, April). Design patterns for code reuse in HLS packet processing pipelines. In 2019 IEEE 27th Annual Int. Symp. on Field-Programmable Custom Computing Machines (FCCM) (pp. 208-217). IEEE.

[19] Mitsuzuka, K., Tokusashi, Y. and Matsutani, H. (2018, December). MultiMQC: A Multilevel Message Queuing Cache Combining In-NIC and In-Kernel Memories. In 2018 Int. Conf. on Field-Programmable Technology (FPT) (pp. 134-141). IEEE.

[20] Tuan, D., Truyen, N., Vu, N., Volvenko, S. and Hung, N. (2019, December). Experimental UWB wireless sensor system for locomotive engine parameters monitoring. In 2019 6th NAFOSTED Conf. on Information and Comp. Sci. (NICS) (pp. 154-159). IEEE.

[21] Sutter, G., Ruiz, M., Lopez-Buedo, S. and Alonso, G. (2018, December). Fpga-based tcp/ip checksum offloading engine for 100 gbps networks. In 2018 Int. Conf. on ReConFigurable Computing and FPGAs (ReConFig) (pp. 1-6). IEEE.

[22] Tokusashi, Y. and Matsutani, H. (2017). Multilevel NoSQL Cache Combining In-NIC and In-Kernel Approaches. IEEE Micro 37(5), 44-51.

[23] Phu, H., Tan, T., Van Men, P., Van Hieu, N. and Van Cuong, T. (2019, December). Design and implementation of configurable convolutional neural network on FPGA. In 2019 6th NAFOSTED Conf. on Information and Computer Science (NICS) (pp. 298-302). IEEE.

[24] Chi, Y., Cui, W., Qiao, L., Long, X. and Wang, L. (2019, March). Hardware Architecture Design of the Deep-learning-based Machine Vision Chip. In 2019 IEEE 3rd Info. Tech., Networking, Electronic and Automation Control Conf. (ITNEC) (pp. 1110-1113). IEEE.

[25] Huang, J., Lou, J., Sun, Y., Wang, T., et al. (2023, October). Making Sense of Using a SmartNIC to Reduce Datacenter Tax from SLO and TCO Perspectives. In 2023 IEEE Int. Symp. on Workload Characterization (IISWC) (pp. 28-42). IEEE.

[26] Huang, J., Lou, J., Vanavasam, S., Kong, X., et al. (2024, June). Hal: Hardware-assisted load balancing for energy-efficient snic-host cooperative computing. In 2024 ACM/IEEE 51st Annual Int. Symp. on Computer Architecture (ISCA) (pp. 613-627). IEEE.

[27] Di Girolamo, S., De Sensi, D., Taranov, K., Malesevic, M., et al. (2022, November). Building blocks for network-accelerated distributed file systems. In SC22: Int. Conf. for High Performance Computing, Networking, Storage and Analysis (pp. 1-14). IEEE.

[28] Basu, S. and Nadig, D. (2024, June). Offloading NVMe over fabrics (NVMe-oF) to SmartNICs on an at-scale distributed testbed. In IEEE 10th Int. Conf. on Network Softwarization (NetSoft) (pp. 316-318). IEEE.

[29] Jain, A., Alnaasan, N., Shafi, A., Subramoni, H. and Panda, D. (2021, August). Accelerating CPU-based distributed DNN training on modern HPC clusters using bluefield-2 DPUs. In 2021 IEEE Symposium on High-Performance Interconnects (HOTI) (pp. 17-24). IEEE.

[30] Jain, A., Alnaasan, N., Shafi, A., Subramoni, H. and Panda, D. (2021). Optimizing distributed DNN Training using CPUs and Bluefield-2 DPUs. IEEE Micro, 42(2), 53-60.
[31] Upadhyay, G., Prasad, C., Akash, V., Singh, A., et al. (2024, August). Enhancing CNN Inference Time and Reducing Latency on Edge and Resource-Constrained Systems through Quantization. In 2024 Second Int. Conf. on Networks, Multimedia and Information Technology (NMITCON) (pp. 1-6). IEEE.
[32] Gongye, C. and Fei, Y. (2024, May). One Flip Away from Chaos: Unraveling Single Points of Failure in Quantized DNNs. Int. Symp. on Hardware Oriented Security and Trust (HOST) (pp. 332-342). IEEE.
[33] Jeong, T., Ghasemi, E., Tuyls, J., Delaye, E. and Sirasao, A. (2020, December). Neural network pruning and hardware acceleration. In IEEE/ACM 13th Int. Conf. on Utility and Cloud Computing (UCC) (pp. 440-445). IEEE.
[34] Vicenzi, J., Korol, G., Jordan, M., Rutzig, M. and Beck, A. (2024, August). Exploiting Virtual Layers and Pruning for FPGA-Based Adaptive Traffic Classification. In 2024 27th Euromicro Conf. on Digital System Design (DSD) (pp. 194-201). IEEE.
[35] Wang, J., Lv, G., Liu, Z. and Yang, X. (2022, September). QUIC encryption offloading based on NanoBPF. In 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS) (pp. 1-4). IEEE.
[36] AMD (n.d.) AMD Vivado Design Suite 2024.2 release now available including major enhancements for designing with AMD Versal ™ adaptive SoCs. https://www.amd.com/en/products/software/adaptive-socs-and-fpgas/vivado.html
[37] Hoefler, T., Di Girolamo, S., Taranov, K., Grant, R. and Brightwell, R. (2017, November). sPIN: High-performance streaming Processing in the Network. In Proceedings of the Int. Conf. for High Performance Computing, Networking, Storage and Analysis (pp. 1-16).

Designing a Plant Leaf Disease Detection Using Image Processing and Machine Learning Techniques

Tokwe Luvolwethu
Department of Computer Science
University of Fort Hare
Alice, South Africa
201927530@ufh.ac.za /
luvolwethutokw1@gmail.com

Nobert R. Jere
Department of Computer Science
University of Fort Hare
Alice, South Africa
njere@ufh.ac.za

Khulumani Sibanda
Department of Applied Informatics
Walter Sisulu University
East London, South Africa
khulumani.sibanda@gmail.com

***Abstract*—This study uses deep learning, primarily convolutional neural networks (CNNs), to create a reliable method for detecting plant leaf diseases. Using a large dataset of over 10,000 high-resolution photographs, the CNN model automatically extracts features from leaf photos and accurately classifies them into disease categories. Experimental results show that this strategy surpasses traditional machine learning techniques, delivering high levels of accuracy and reliability. The performance of the CNN-based system demonstrates its potential for enabling fast disease identification, minimizing dependency on human inspection, and boosting sustainable agricultural practices by facilitating early and precise intervention.**

Keywords— Machine Learning, Image Processing, Keras, TensorFlow, Detection, Accuracy, Plan Village Dataset, Kaggle, Model.

I. INTRODUCTION

Emerging technologies such as AI and computer vision are being used in industries such as healthcare and finance to speed up operations and reduce human error. This research focuses on their implementation in agriculture, a sector critical to global economic growth, poverty eradication, and food security, particularly in regions such as Southern Africa, where more than 70% of the population relies on it [1].

Rising demand from a growing global population, combined with substantial obstacles such as crop diseases, place a significant strain on the agricultural system's output and economic performance [1]. Current technologies for plant disease identification, which use machine vision to evaluate photos, are challenged by numerous obstacles. These include a scarcity of resources for improved detection technologies and farmers' inadequate ability to identify diseases manually. Crucially, current systems frequently fail to give early-stage predictions, which are required for effective intervention [2]. This gap highlights the vital need of providing accessible technology. Early diagnosis enables farmers to treat plants quickly, minimizing extensive damage and crop loss, and thereby improving food security and economic stability [3]. To solve this, the researchers suggest a smart farming system that uses image processing and machine learning to detect illnesses in leaf images. This strategy seeks to duplicate the success of previous research, such as a system that identified diseases with 93% accuracy, but with an emphasis on overcoming resource restrictions [3].

Unlike many existing solutions that rely on data from urban farms or expensive drones, this research aims to develop a system that requires fewer resources. By efficiently extracting features from images and using machine learning to automate detection, the proposed technology promises to provide accurate, fast, and easy-to-use disease identification. This will empower farmers to make better farming decisions, ultimately supporting sustainable agricultural practices and improving yields with minimal investment [4].

II. BACKGROUND

Early methods of diagnosing plant diseases, such as naked-eye inspection, were notoriously unreliable. They were costly, time-consuming, and required highly skilled experts, posing a considerable barrier to effective crop protection [5]. While modern technologies such as drones and aerial photography are being studied to automate detection, these approaches are frequently resource-intensive and unsustainable for general usage [5].

The use of Convolutional Neural Networks (CNNs) has been a significant step forward in overcoming these challenges. Unlike previous methods, CNNs automate the feature extraction process, allowing for highly accurate and quick analysis of leaf pictures without requiring ongoing human intervention [6]. Pre-trained architectures such as ResNet-50, VGG-16, and DenseNet-121 have been effectively used in studies, indicating their greater capacity to identify diseases precisely [7], [8]. The use of these deep learning models is critical because they provide a scalable, economical, and resilient solution that outperforms manual inspection and other technical methods [8].

As a result, relying on this backdrop, which emphasizes the strain of early growth and the transition toward AI, this study seeks to employ these very image processing and machine learning approaches. The goal is to create a more accessible and accurate plant disease detection system that directly addresses the historical obstacles identified in previous studies.

III. PROBLEM STATEMENT

The fact that existing plant disease detection systems are crop-specific is a significant constraint. While many solutions have been developed using traditional machine learning approaches, they are often tailored to a specific plant, such as apples [10], [12]. This presents a huge challenge for the large majority of farmers that practice multi-cropping, as it would necessitate a different, specialized system for each crop, which is inefficient as well as expensive [11].

Convolutional Neural Networks (CNNs) provide an excellent answer to this challenge. Their deep learning approach is particularly effective at automatically learning generalized features from visual input. This inherent capacity enables a single, well-designed CNN model to be trained on a varied dataset that includes multiple plant species [11]. As a result, CNNs can deliver effective disease diagnosis across multiple crops using a single unified system, thereby addressing the inflexibility and limitations of existing crop-specific machine learning systems [11].

IV. Aim, Question and Objective

The major goal of this research is to create a smart agricultural system that uses a Convolutional Neural Network (CNN) to enable accurate, early disease diagnosis for farmers who lack access to agricultural professionals [11]. This immediately tackles the serious shortage of extension officers, especially in South Africa's Eastern Cape.

By using the CNN's better ability to automatically extract complex features from images, the system will outperform traditional machine learning techniques, which sometimes involve manual feature engineering and are less versatile [11]. The ultimate result will be a robust, user-friendly application that successfully bridges the diagnostic gap, allowing farmers to undertake speedy detection and treatment, and thereby boosting precision agriculture through accessible and scalable AI-driven solutions [12].

The main question is how can machine learning and image processing be used to create a smart agricultural disease detection system? To answer this, an objective is to design a system, with the use of image processing and machine learning algorithms, specifically CNNs, that will detect diseases from the leaves of plants and classify these diseases.

Classical machine learning (ML) has played an important role in automating plant disease identification. These professional systems work in two steps: first, specialists manually detect and extract relevant information from leaf photos, such as color histograms, texture patterns (e.g., from GLCM), and shape descriptors. These manufactured traits are then used to train classifiers, such as Support Vector Machines (SVMs) or Random Forests, to differentiate between healthy and unhealthy plants [12]. This method has been successful in diagnosing specific diseases such as powdery mildew, rust, and leaf blotch, as well as distinguishing disease symptoms from abiotic pressures such as nutrient deficit [12].

However, the efficacy of these traditional ML techniques is fundamentally constrained because they rely on human feature extraction. This procedure is not only time-consuming and necessitates specialist subject knowledge, but it also fails to capture the complex, subtle, and frequently non-linear traits that distinguish early-stage infections [13]. As a result, these models frequently lack the precision required for accurate early detection and may fail to generalize across various lighting conditions, plant kinds, or disease strains[14].

In contrast, Convolutional Neural Networks (CNNs) represent a paradigm change by automating feature extraction and categorization inside a single, end-to-end learnable architecture. Their superiority originates from their deep, hierarchical structure, which allows them to automatically acquire progressively complicated feature representations—from simple edges and colors in early levels to intricate textures and disease-specific patterns in later layers. This intrinsic aptitude enables CNNs to excel at detecting subtle, early-stage symptoms that are frequently unnoticeable to manually constructed features, resulting in much improved accuracy, robustness, and generalizability, setting a new standard in the area [11], [15].

Convolutional Neural Networks (CNNs) in plant disease detection systems benefit greatly from image processing integration. The first steps entail preprocessing the acquired leaf image using techniques such as scaling, normalization, and noise reduction. This standardization is critical because it provides a clean, consistent input for the CNN, ensuring that the model concentrates on key features rather than artifacts or differences in lighting and background [15].

Following that, segmentation is used to separate the sick area (the genuine region of interest) from the healthy leaf tissue and background. This exact localization is critical because it focuses the CNN's computational resources to analyze the symptomatic area, significantly increasing the model's efficiency and diagnostic accuracy [14].

Traditional systems required manual feature extraction at this step, whereas a CNN automates the process. The preprocessed and segmented image is fed directly into the network, which self-learns to extract the most discriminative hierarchical features for classification [15]. Thus, image processing serves as an important preprocessing pipeline, optimizing raw data to fully use the CNN's powerful, automatic feature learning capabilities, resulting in a more robust and accurate diagnosis [15].

V. Related Literature

The development of plant leaf disease detection from The transition from conventional diagnostic techniques to Convolutional Neural Networks (CNNs) signifies a paradigm change towards automation, accuracy, and scalability, directly addressing the significant shortcomings of its predecessors, according to a comprehensive review of the literature. This shift represents a fundamental reimagining of how plant disease detection is performed, moving away from subjective human analysis to data-driven, algorithmic precision.

Traditional diagnosis procedures, which rely on manual visual inspection by experts [17] or laboratory-based biological and mycological studies, have inherent limitations. These methods are not only time-consuming but also necessitate specialized expertise, which is frequently rare in resource-limited locations such as South Africa's Eastern Cape. Furthermore, they struggle significantly to detect early-stage or non-targeted diseases [16], often identifying an issue only after it has already established itself and become more difficult to manage.

The earliest computational approaches that followed relied on classical machine learning, which, while improving over purely manual methods, created a new set of restrictions. According to [16], these systems relied on a cumbersome multi-stage pipeline that included image pre-processing, diseased area

segmentation, and manual feature extraction with a focus on engineered characteristics like color, texture, and shape. Classification was then performed using models such as Support Vector Machines [18] or k-means clustering [17]. The main disadvantage of this strategy was its critical reliance on manual feature engineering, where the overall system's accuracy was entirely dependent on the developers' ability to discover and extract the most relevant information.

Convolutional Neural Networks (CNNs) have overtaken these traditional machine learning techniques by automating the most crucial and limiting step: feature extraction. CNNs, such as those used by [18] and [21], build hierarchical representations of features directly from raw pixel data using their deep, layered architectures. Initial layers learn basic elements like edges and colors, whereas deeper layers combine them to create more complicated patterns and textures particular to diseases. This end-to-end learning eliminates the need for manual feature engineering, allowing the model to independently learn the most discriminative characteristics for classification.

The literature reveals that this architectural advantage leads to concrete, superior results. Studies routinely show accuracy rates more than 90%, a significant improvement over traditional methods. For example, [22] obtained 92.23% accuracy with a CNN, whereas [20] reported above 99% accuracy using deep learning models. The scalability of CNNs is demonstrated by research such as that of [23], who created a system capable of detecting 20 illnesses across five plant species with 93% accuracy, demonstrating the multi-class capabilities that provide a clear solution to the limitations of plant-specific models.

Furthermore, techniques such as transfer learning [20] have significantly advanced the subject by mitigating the challenge of large, annotated datasets. Researchers can achieve high accuracy on often-limited agricultural datasets by fine-tuning pre-trained networks (e.g., VGGNet, ResNet), which have already learned rich feature hierarchies from large corpora like ImageNet. The progress continues with advanced hybrid models that combine CNNs' powerful automated feature extraction with the robust classification capabilities of SVMs [18], pushing the performance bounds even further.

This technological evolution has profound implications beyond mere accuracy. The development of highly accurate, automated, and scalable deep learning solutions is crucial for global food security, as it provides a proactive and precise tool for disease management. By enabling early detection, this technology can help reduce economic losses for farmers and decrease reliance on limited expert knowledge, making advanced diagnostic tools accessible in a wider range of agricultural settings.

Finally, the literature clearly demonstrates an evolutionary path: from slow, expert-dependent manual inspection to partially automated but limited traditional machine learning, and finally to the current state of deep learning solutions driven by CNNs. CNNs do more than just improve on previous methods; they fundamentally alter the approach by automating feature discovery, resulting in unprecedented accuracy, early detection, and the development of versatile systems capable of meeting the complex needs of modern, multi-crop agriculture.

VI. DESIGN AND METHODOLOGY

This research was undertaken to fill a critical gap in agricultural technology: the absence of a dynamic, accurate, and multi-crop plant leaf disease detection system. A thorough analysis of over 1300 papers found that, while traditional machine learning (ML) approaches have been investigated, their constraints prevent practical use [19]. This resulted in the use of a Convolutional Neural Network (CNN)-based technique, which was created, trained, and tested to provide a reliable prediction system. The study began by assessing classic ML approaches. Support Vector Machines (SVM) were eliminated because to their high computational cost and lengthy training durations on huge picture datasets. Random Forest algorithms were also rejected because their ensemble structure, which includes several decision trees, makes them too sluggish and ineffectual for real-time predictions, which is an essential criterion for a field-deployable tool. Transfer Learning was examined; however it carried the danger of overfitting on our unique dataset and negative transfer, in which non-optimal pre-learned features degraded performance [24].

The CNN architecture was selected as the superior alternative for several key reasons:

- **Automatic Feature Extraction**: Unlike SVM and Random Forest, which require manual, time-consuming, and often subjective feature engineering (e.g., extracting texture, colour, and shape metrics), CNNs automatically and hierarchically learn the most discriminative features directly from the raw pixel data. This integration of feature extraction and classification into a single, end-to-end trainable model is a fundamental advantage [22].
- **Spatial Hierarchical Learning**: CNNs use convolutional layers with filters that scan the input image, learning to detect edges, textures, and eventually complex patterns like disease-specific lesions. This spatial hierarchy makes them inherently suited for image data, a capability traditional ML models lack [24].
- **Computational Efficiency for Inference**: Once trained, a CNN can analyse fresh pictures using a sequence of optimized matrix operations, making it extremely efficient for real-time prediction, outperforming slower ensemble approaches [23].

The model was created using Kaggle's PlantVillage dataset, which contains over 8,000 high-quality photos of healthy and sick leaves from three important crops (tomatoes, potatoes, and grapes) organized into six groups. To prepare data for CNN:

- **Resizing**: Each image was scaled to 128x128 pixels. This standardization is a fundamental input requirement for CNNs, and it considerably reduces computational effort while preserving key properties.
- **Normalization**: involved scaling pixel values to a range of [0, 1] using min-max scaling. This normalization stabilizes the training process by

maintaining constant feature scales, which results in faster convergence and higher model accuracy.

- **Augmentation (Training Set Only)**: The training dataset was artificially extended by techniques like as rotation, flipping, and scaling. This augmentation greatly increases the variability of the training data, teaching the CNN to be insensitive to these differences and significantly improves its capacity to generalize to new, previously unseen pictures.

The CNN was trained using 80% of the data, with 15% for validation and 5% for final testing.

- **Optimizer:** The Adam optimizer was chosen for its adjustable learning rate capabilities, which combine the benefits of AdaGrad and RMSProp to ensure efficient and reliable convergence.
- **Learning Rate:** An initial learning rate of 1e-5 was set to provide for regulated, accurate weight updates while limiting divergence and assuring consistent learning.
- **Loss Function:** Categorical Cross-Entropy was utilized as the loss function, which is appropriate for multi-class classification since it penalizes wrong predictions and drives the model to high classification confidence.
- **Training Regime:** The model was trained for 10 epochs with a batch size of 32, achieving a compromise between computational efficiency and consistent learning.

Accuracy and loss curves were used to rigorously track performance on both training and validation sets. Initial iterations with a big, complicated dataset (13 classes) resulted in low accuracy (~32%), illustrating the problem of model complexity. The dataset was adjusted iteratively to three crops and six classes, allowing the CNN to train successfully, reaching a test accuracy of 56% and demonstrating the importance of data amount and quality on CNN performance. The confusion matrix and a comprehensive classification report (precision, recall, and F1-score) were examined to identify particular classes in which the model misidentified illnesses, so offering obvious paths for future model improvement.

The final trained CNN model was included into a prediction system. The workflow includes:

- **Input:** A user uploads a leaf picture.
- **Preprocessing**: The image is automatically resized and normalized to 128x128 pixels.
- **Prediction**: The picture is passed into a CNN, which generates a probability distribution across six classifications.
- **Output:** The class with the highest probability is chosen. The algorithm then obtains a full description of the diagnosed ailment from a prepared JSON database, displaying the input image with the prediction and description for easy understanding.

This study shows that CNNs are a far more powerful framework for image-based plant disease identification than typical machine learning approaches. Their capacity to execute autonomous, hierarchical feature extraction removes the bottleneck of manual engineering, resulting in higher accuracy [25]. The inclusion of image processing stages such as scaling and normalization straight into the CNN pipeline results in a robust, end-to-end system that can learn directly from pixels. While problems such as dataset scope and model complexity arose, the CNN architecture demonstrated its durability and iterative improvement capabilities, laying the groundwork for the development of a viable, multi-crop disease detection tool [25].

VII. RESULTS

The iterative creation and assessment of the Convolutional Neural Network (CNN) model produced many discoveries. The model's performance was largely evaluated using accuracy, training length, and confusion matrix analysis, with the results averaged for easier comparison.

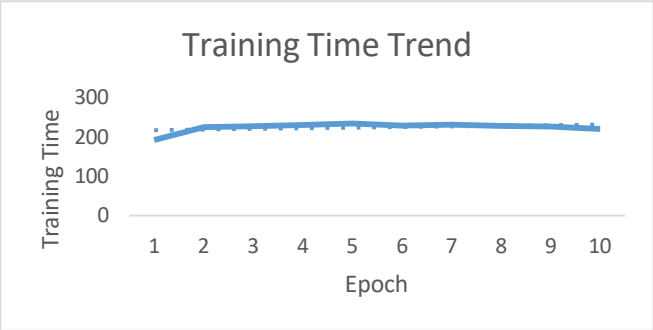
A major outcome was the model's final average accuracy of 56%, which was a huge improvement over the original 32%. The performance was rated good since it exceeded the experiment's 50% accuracy requirement. Crucially, the CNN design revealed a clear trend: accuracy increased inversely with data amount, from 32% with 9 classes to 56% with a more concentrated 6-class sample. This demonstrates a significant benefit of CNNs, their capacity to perform with less but more relevant data by decreasing background noise and focusing on the most discriminative characteristics inside pictures, a task that traditional machine learning approaches, which rely on human feature extraction, frequently fail. Table I represents the accuracies that the model has obtained during the training process.

TABLE I. MODEL ACCURACIES DURING TRAINING

Model Accuracies During Training		
Epoch	***Training***	***Validation***
Epoch 1/10	54.58%	55.85%
Epoch 2/10	56.41%	55.85%
Epoch 3/10	56.08%	55.85%
Epoch 4/10	56.28%	55.85%
Epoch 5/10	56.15%	55.85%
Epoch 6/10	55.49%	55.85%
Epoch 7/10	56.34%	55.85%
Epoch 8/10	56.15%	55.85%
Epoch 9/10	56.15%	55.85%
Epoch 10/10	56.19%	55.85%
Average	55.98%	55.85%

[a.] Model Accuracies During Training

The training procedure, which took an average of 224 seconds each epoch, exhibited varying durations, reflecting the model's adaptive learning process across repetitions. The graph below demonstrates the training time trend, it clearly shows that during the first epochs the training time is below the equilibrium, however as the model trains it adopts the spatial characters and it then goes above the equilibrium showing the good results in the training.



The Training Time Trend

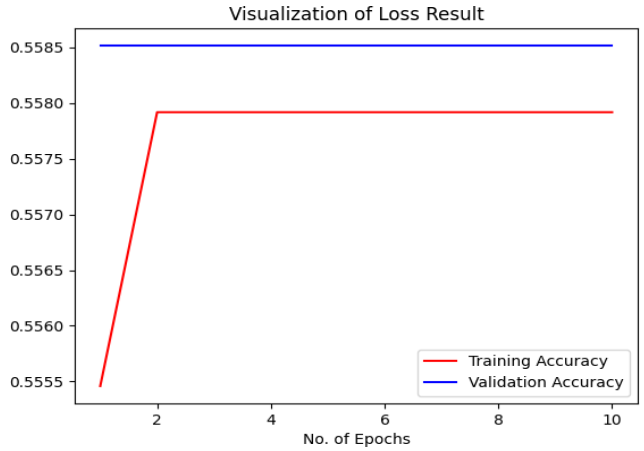
Table II demonstrates the loss function that is obtained during the training process.

TABLE II. MODEL LOSS FUNCTION DURING TRAINING

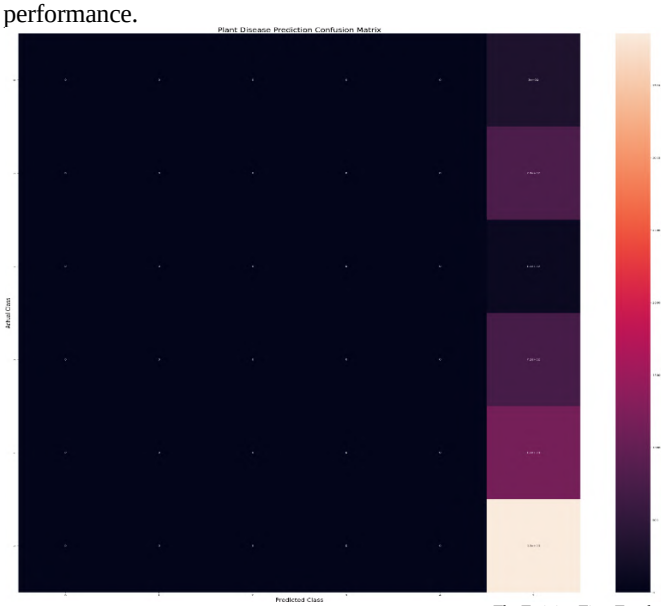
Model Loss Function During Training		
Epoch	***Training***	***Validation***
Epoch 1/10	14.93%	14.72%
Epoch 2/10	14.53%	14.72%
Epoch 3/10	14.64%	14.72%
Epoch 4/10	14.57%	14.72%
Epoch 5/10	14.62%	14.72%
Epoch 6/10	14.69%	14.72%
Epoch 7/10	14.55%	14.72%
Epoch 8/10	14.62%	14.72%
Epoch 9/10	14.62%	14.72%
Epoch 10/10	14.60%	14.72%
Average	14.64%	14.72%

Model Loss Function During Training

The loss function values, which measure the difference between predictions and actual outcomes, stabilized at an average of 14.64%, suggesting continuous learning convergence over epochs. The figure below demonstrates the trend between the training accuracy against the validation accuracy.



Furthermore, the confusion matrix and subsequent classification reports provided a granular view of the model's performance.



The Training Time Trend

These demonstrated the CNN's ability to correctly identify crop disease classes while also identifying its misclassification trends. Metrics like accuracy, recall, and F1-score provided a fair view of its strengths and limitations per class. This extensive diagnostic capacity is a direct result of CNNs, which automatically learn hierarchical feature representations (e.g., edges, textures, patterns) that are significantly more effective for complicated picture categorization than hand-crafted features utilized in older approaches. This fundamental characteristic allowed the model to attain its claimed effectiveness in distinguishing various crop diseases.

VIII. DISCUSSION AND CNN EFFICACY

The results found that the model performed substantially better on the segmented dataset, with greater accuracy. This improvement is mostly due to the strengths of the Convolutional Neural Network (CNN) architecture. Unlike

classic machine learning models, which involve laborious and frequently rudimentary feature extraction (for example, shape or color histograms), CNNs learn hierarchical and discriminative features directly from pixel data. The segmented dataset reduced background noise and computational burden, allowing CNNs to focus its learning capacity on the most significant patterns and textures particular to each illness, improving predicted precision.

This efficiency was further enhanced by hardware capabilities, as CNNs are computationally demanding and need strong processors. While the model took longer to train at first, its learning efficiency increased with each epoch, thanks to the CNN's backpropagation method, which iteratively refines internal filters.

However, certain problems persisted. Misclassification of visually identical illnesses was a major concern, which would be exacerbated in previous models due to their limited feature detection capabilities. While the CNN considerably improved diagnostic potential, it also demonstrated that a purely visual method had inherent limitations for discerning tiny morphological abnormalities. Furthermore, class imbalance was a concern, since underrepresented classes obtained less robust feature learning. However, the CNN architecture gave the ability to solve this using integrated approaches such as data augmentation and class weight modification, which are less effective or more difficult to execute in typical pipelines.

Finally, the model's performance under controlled settings may not be entirely applicable to real-world field applications with changeable lighting and backdrops. Nonetheless, CNN's capacity to learn generalized, invariant properties makes it more resistant to such fluctuations than older approaches, laying a solid platform for future real-world implementation. To summarize, while obstacles persist, the CNN architecture proved critical in attaining significant accuracy increases and provides a robust, customizable framework for overcoming the inherent constraints of image-based crop disease detection.

IX. CONCLUSION

This study shows that Convolutional Neural Networks (CNNs) may significantly improve plant leaf disease detection. The CNN architecture was critical to the system's performance because of its inherent capacity to automatically build hierarchical and discriminative features from raw pixel data, which considerably outperforms typical machine learning approaches that rely on human feature extraction. This proved critical for correctly recognizing complicated visual patterns linked with numerous crop diseases.

While obstacles such as identifying visually similar illnesses and regulating class imbalance were identified, the CNN architecture provided a robust and adaptable platform for addressing these concerns using approaches such as data augmentation and transfer learning. The model's performance, particularly on segmented datasets, demonstrates CNNs' usefulness in reducing noise and focusing on important characteristics.

Future development could build on the CNN's strengths by increasing datasets with more diverse pictures and including multimodal data, such as environmental sensors, to produce a more powerful diagnostic tool. The ultimate objective is to deploy a CNN-powered system in the real world, maybe via a smartphone application, to provide farmers with an accessible and accurate method of early disease diagnosis. To summarize, CNNs have shown to be a breakthrough technology for agricultural AI, providing a fundamental basis for designing systems that contribute to increased crop health, yields, and more sustainable farming techniques.

X. FUTURE WORKS

Future research into optimizing Convolutional Neural Network (CNN)-based plant disease detection systems should concentrate on improving data, model design, and integration. A main objective is to considerably extend and diversity the training dataset. Curating a wider collection of photographs, particularly those addressing underrepresented illnesses and class disparities, is critical. This data should be collected under a variety of real-world situations, including variable lighting, angles, and backdrops, to improve the model's resilience and flexibility, giving it a significant edge over classic ML algorithms that struggle with such diversity.

To improve CNN performance, improved preprocessing and innovative data formats should be combined. Sophisticated picture enhancement and noise reduction techniques can help to sharpen disease-specific characteristics. More importantly, adding hyperspectral or multispectral imaging would enable CNNs to detect subtle, pre-visual illness indications, allowing for earlier and more accurate diagnosis than normal RGB pictures and manual feature engineering alone.

The model itself can be improved by architectural changes. Exploring hybrid models that combine CNN feature extraction with sophisticated classifiers such as Support Vector Machines (SVM) may improve accuracy, especially when discriminating between similar illnesses. Furthermore, including attention mechanisms or transformer models would enable the network to dynamically focus on the most relevant visual areas, resulting in increased accuracy.

Finally, going beyond plain picture analysis is critical. Optimizing the system entails developing a multimodal framework that combines CNN-extracted visual characteristics with contextual information such as soil health, temperature, and humidity. This comprehensive technique provides not just detection but also predictive insights into disease outbreaks. Validating these enhanced CNNs via rigorous field testing and implementing them via user-friendly mobile applications will demonstrate their superiority to old approaches, providing a scalable, proactive, and precise tool for sustainable agriculture.

REFERENCES

[1] Ainembabazi, J. H., van Asten, P., Vanlauwe, B., Ouma, E., Blomme, G., Birachi, E. A., Nguezet, P. M. D., Mignouna, D. B. and Manyong, V. M. (2017). "Improving the speed of adoption of agricultural technologies and farm performance through farmer groups: evidence from the Great Lakes region of Africa". Agricultural Economics 48(2): 241–259.

[2] Almeida, Fausto et al.(2019) "The Still Underestimated Problem of Fungal Diseases Worldwide." Frontiers in microbiology vol. 10 214.

[3] Andria E, Amanda R, (2023), "Types, Causes, and Prevention of Crop Diseases. The Homestead Survival". https://eos.com/blog/crop-diseases/

[4] Atila, Ü., Uçar, M., Akyol, K., & Uçar, E. (2021). Plant leaf disease classification using EfficientNet deep learning model. Ecological Informatics, 61, 101182.

[5] Awuor, F., Otanga, S., Kimeli, V., Rambim, D. and Abuya, T. (2019). "e-Pest Surveillance: large scale crop pest surveillance and control". 2019 IST-Africa Week Conference (IST-Africa). IEEE, 1–8.

[6] Bin L. (2005) "Key Laboratory of Molecular Biology of Crop Pathogens and Insects", Institute of Biotechnology, Zhejiang University, Hangzhou 310058, China

[7] Brown, L., & Green, M. (2022). "Handling Class Imbalance in Machine Learning." International Journal of Data Science.

[8] Brown, L., & Green, P. (2022). Fine-Tuning Models for Accurate Disease Prediction. International Conference on Machine Learning, 34(1), 456-467.

[9] Calicioglu, O., Flammini, A., Bracco, S., Bellù, L., & Sims, R. (2019). The Future Challenges of Food and Agriculture: An Integrated Analysis of Trends and Solutions. Sustainability, 11(1), 222. https://doi.org/10.3390/su11010222

[10] Connor S, Taghi K and Borko F. (2021) "Text Data Augmentation for Deep Learning". Journal of Big Data volume 8, Article number: 101.

[11] Davidian, M., Lahav, A., Joshua, B.-Z., Wand, O., Lurie, Y., & Mark, S. (2024). "Exploring the Interplay of Dataset Size and Imbalance on CNN Performance in Healthcare. Diagnostics", 14(16), 1727.

[12] Demilie, W.B. (2024). "Plant disease detection and classification techniques: a comparative study of the performances". Journal of Big Data 11, 1. DOI:10.1186/s40537-023-00863-9

[13] Deutsch, C. A., Tewksbury, J. J., Tigchelaar, M., Battisti, D. S., Merrill, S. C., Huey, R. B., & Naylor, R. L. (2018). CLIMATE CHANGE Increase in crop losses to insect pests in a warming climate. In Science (Vol. 361). https://www.science.org

[14] Dodds, P.N.; Rathjen, J.P. (2010) Plant immunity: Toward an integrated view of plant–pathogen interactions. Nat. Rev. Genet, 11, 539–548.

[15] Domingues, T., Brandão, T., & Ferreira, J. C. (2022). Machine Learning for Detection and Prediction of Crop Diseases and Pests: A Comprehensive Survey. In Agriculture (Switzerland) (Vol. 12, Issue 9). MDPI. https://doi.org/10.3390/agriculture12091350

[16] Dubey S. R, and Jalal A. S. (2012). "Detection and Classification of Apple Fruit Diseases Using Complete Local Binary Patterns," IEEE Computer and Communication Technology (ICCCT), pp. 346–351.

[17] Esker, PA, D. (2023) Improved integrated disease management based on science-based, actionable information obtained at different scales. USDA School of Grassland Science, Beijing Forestry University, Beijing 100083, China

[18] Farzand A., Moosa A., Zubair M., Khan A.R., Massawe V.C., Tahir H.A.S., Sheikh T.M.M., Ayaz M., Gao X.(2019). "Suppression of Sclerotinia sclerotiorum by the Induction of Systemic Resistance and Regulation of Antioxidant Pathways in Tomato Using Fengycin Produced by Bacillus amyloliquefaciens FZB42. Biomolecules". PMID: 31623124 PMCID: PMC6843208 DOI: 10.3390/biom9100613

[19] Fróna, D., Szenderák, J., & Harangi-Rákos, M. (2019). The challenge of feeding theworld. In Sustainability (Switzerland) (Vol. 11, Issue 20). MDPI. https://doi.org/10.3390/su11205816

[20] Gala, D., Khetan, S., & Mehendale, N. (2023). Revolutionizing Agriculture: Disease Detection in Crops with Deep Learning and Drone Imagery. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4665793

[21] Hannah, R., Lucas, R.-G., Edouard, M., Marcel, G., Esteban, O.-O., Joe, H., & Max, R. (2023). Population Growth. Our World In Data. https://ourworldindata.org/population-growth

[22] Hughes, D., & Salathé, M. (2015). An open-access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060.

[23] Irina P, and Phoey L. T, (2018). "Text Segmentation - Approaches, Datasets, and Metrics". Studies in Computational Intelligence 741:167-181. DOI:10.1007/978-3-319-66984-7_10

[24] Jafar, A., Bibi, N., Naqvi, R. A., Sadeghi-Niaraki, A., & Jeong, D. (2024). Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations. In Frontiers in Plant Science (Vol. 15). Frontiers Media SA. https://doi.org/10.3389/fpls.2024.1356260

[25] Jones, J.D.; Dangl, J.L. The plant immune system. Nature 2006, 444, 323–329.

Developing a Large Language Model for Text Generation in the Under-Resourced Sepedi language.

1st Mathews Tladi
Department of Computer Science
University of Limpopo
Polokwane, South Africa
matthewstladi22@gmail.com

2nd Madimetja Jonas Manamela
Department of Computer Science
University of Limpopo
Polokwane, South Africa
jonas.manamela@ul.ac.za

3rd Thipe Modipa
Department of Computer Science
University of Limpopo
Polokwane, South Africa
thipe.modipa@ul.ac.za

***Abstract*—The rapid advancement of Large Language Models (LLMs) has significantly improved natural language processing (NLP) tasks such as text generation. High-resourced languages like English have largely benefited from these developments, leaving under-resourced languages like Sepedi underrepresented. This research proposes the development of a transformer-based LLM tailored for Sepedi question-answering text generation. This paper highlights and addresses challenges such as data scarcity, limited computational resources, and high training costs. The paper focuses on compiling and curating a large, high-quality Sepedi corpus from diverse sources. The training strategy adopted the transformer architecture DistilGPT-2 that was applied on a small scale. The model's performance was evaluated using perplexity, BLEU, and ROUGE. The results show BLEU [0.40], ROUGE-1 [0.21], ROUGE-L [0.19], and perplexity [19], thereby indicating an overall moderate performance of the model. Although optimal results were not achieved in this preliminary phase due to storage and computational constraints in Google Colab, a baseline Sepedi text-generation LLM was successfully developed.**

Keywords: large language model, text generation, transformer, low-resource language.

I. INTRODUCTION

Artificial intelligence (AI) has made significant speech and language processing advancements recently [1]. This is especially evident in the advancement of the AI-driven language models that are intended for text generation. A language model in natural language processing (NLP) refers to a statistical or machine learning approach that can predict the next word in a sequence of text, given the context of the previously used words [2]. Large language models (LLMs) are AI-based models that use neural networks with a million to a billion sets of parameters to analyse, comprehend, and produce text, typically through transformer-based deep learning architectures [3]. However, the development of these models has primarily focused on highly resourced languages such as English, as a result leaving most of the low-resourced languages such as the Sepedi language, less technologically developed.

Text generation, or natural language generation, is a core NLP task that produces fluent and coherent text from input data [4]. Applications of text generation span across key areas such as machine translation [5] and summarisation [6]. The recent advancements in deep learning, particularly in recurrent, convolutional, and attention-driven architectures, were shown to have greatly improved the end-to-end modelling of semantic mappings [4]. Nevertheless, these advances are often hindered by the scarcity of large-scale supervised datasets, which increases the risk of overfitting and constrains generalization. Despite these limitations, text generation continues to play a central role in NLP research, driving innovations in natural language understanding and powering diverse real-world applications [7].

According to Wong [8], the dominance of the English language in AI-based digital technologies for speech and language processing contributes to the decline in the use and appeal of many low-resourced languages. The inclusive development of speech and language processing digital platforms in large language modelling is essential to prevent the cultural and linguistic erosion in low-resourced languages [9]. Research efforts, such as those of Alabi et al. [10], have shown the potential of transformer-based models for African languages. However, the Sepedi language remains under-represented in such language development works. While Masethe et al. [11] have explored paraphrase generation in the Sepedi language, the broader landscape of NLP research continues to marginalise the language's unique linguistic and computational requirements, thereby highlighting the need for a more comprehensive and inclusive approach. This lack of adequate digital representation for the Sepedi language hinders linguistic inclusivity, cultural preservation, and equal access to NLP technologies amongst the South African low-resourced languages. Developing an LLM for Sepedi poses several challenges that include the scarcity of text-based training data, limited computational resources, and high training costs [12], [13].

This study aims to develop a transformer-based large language model for text generation in the Sepedi language. The objectives of the research include: (1) Collect and curate a massive text dataset that is adequate to develop a text-generating LLM for the Sepedi language; (2) Utilize the Sepedi language text data to train a transformer-based large language model for text generation; (3) Evaluate the quality of the developed large language model with relevant metrics such as perplexity, the Bilingual Evaluation Understudy, and the Recall-Oriented Understudy for Gisting Evaluation.

The rest of the paper is outlined as follows. Section II provides a broad overview of related LLM research, the methodology is presented in Section III, Section IV presents the preliminary results, and Section V outlines the conclusion of the study.

II. LITERATURE REVIEW

Most prior text generation research work conducted globally in the domain of innovative NLP has been directed towards the well-resourced languages like English, Spanish, and Mandarin, therefore resulting in the availability of large datasets and extensive computational resources for these languages [14]. In contrast, African languages continue to face a significant research gap due to the lack of adequate, curated

text corpora, which poses a major challenge in training robust LLMs for African indigenous underrepresented languages [15].

Recognising this development disparity, Conneau et al. [16] initiated efforts to support the low-resourced languages through a transfer learning approach. These were leveraged by multilingual models such as the Multilingual Bidirectional Encoder Representation from Transformers (mBERT) and the Cross-Lingual Language Model based on a Robustly Optimised BERT Pretraining Approach (XLM-RoBERTa) to improve the adaptation to these low-resourced languages. Recent initiatives to address the research gap in African NLP have begun by promoting the creation of open, high-quality datasets and encouraging collaboration of researchers across the African continent [17]. Notably, the MasakhaNER project [18] created the first large-scale, publicly accessible named entity recognition (NER) dataset covering ten African languages, which supports both supervised and transfer learning approaches, enabling a deeper exploration of African NLP and its linguistic diversity.

Beyond African languages, a study conducted by Ehsan and Solorio [19] demonstrated how data augmentation can enhance NER performance in the low-resourced Pakistani languages, namely, Shahmukhi and Pashto. This was achieved by leveraging cross-lingual data augmentation and fine-tuning existing multilingual large language models, significantly improving NER outcomes for these underrepresented languages. These findings align with Chen et al. [20], who conducted an empirical study on the effectiveness of various data augmentation methods, including token-level and sentence-level approaches, on the performance of large language models in low-resource settings.

The Transformers, originally introduced by Vaswani et al. [21], are deep learning architectures that have become the backbone of most modern NLP systems. These models use a self-attention mechanism to process input sequences in parallel, allowing them to capture contextual relationships more effectively than previous sequential models such as recurrent neural networks (RNNs) or long short-term memory networks (LSTMs). Fig. 1 illustrates the transformer model architecture, highlighting its encoder and decoder components that underpin these advancements.

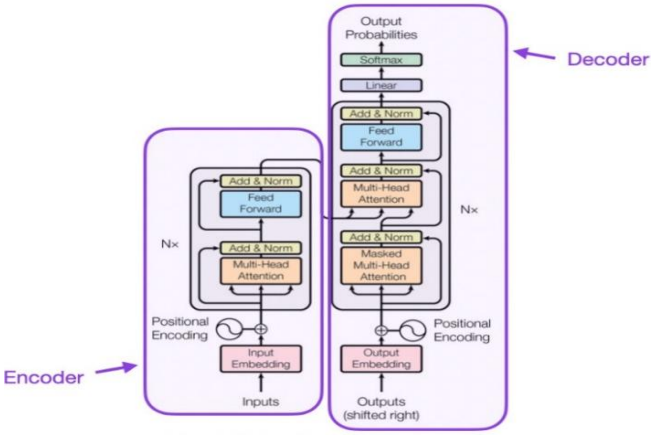


Fig. 1. The transformer architecture model [21]

Building on this foundation, researchers have developed increasingly large and sophisticated transformer-based models. Notable examples include Bidirectional Encoder Representations from Transformers (BERT), Generative Pretrained Transformers (GPT, GPT-2, GPT-3 and GPT-4), and XLM-R, which were made possible through advancements in model scaling, optimisation techniques, and computational infrastructure [22]. A study conducted by Liu et al. [23] adopted a direct approach to addressing low-resourced language challenges by utilising the Multilingual Bidirectional Autoregressive Transformer (mBART) framework, which demonstrated notable effectiveness in such contexts.

The encoder transforms the input tokens into embeddings, incorporating a positional encoding to capture sequential information, and processes these representations through stacked layers of multi-head attention and feed-forward networks. The decoder adopts a similar structure; however, it additionally integrates masked multi-head attention to ensure autoregressive generation, thereby predicting each token step by step without access to future information. Once this process is complete, the decoder produces probability distributions over the vocabulary through a SoftMax layer, enabling the generation of coherent and contextually appropriate text. The architecture has been shown to effectively capture long-range dependencies and has become foundational in modern NLP research [21], [24].

Although the transformer-based LLM architectures have demonstrated impressive success in tasks such as text generation and question answering, their performance relies heavily on the availability of large-scale, clean text datasets [25]. This reliance implies they may be less effective for low-resourced languages where such data is scarce. To address data scarcity in low-resourced languages, researchers have explored various text collection methods such as web scraping, crowdsourcing, and digitizing existing literature to augment the volume of training data [26]. Despite these well-meaning and promising efforts, there is still a need for more research focused specifically on developing LLMs for indigenous South African languages such as Sepedi [27]. Studies by Nekoto et al. [15] and Adelani et al. [18] have further emphasized that increasing the availability of digital resources is crucial for building a pioneering LLM for text generation in the under-resourced Sepedi language, with the goal of promoting and diversifying digital inclusion and linguistic equity.

To support LLM development in low-resourced settings, the Distilled Generative Pretrained Transformer serves as a compact Transformer baseline. Its reduced size lowers compute and memory demands, thereby enabling experimentation in limited resource setting [28]. Additionally, the Low-Rank Adaptation (LoRA) offers a parameter-efficient fine-tuning by learning compact low-rank updates, which preserve the base model [29] and transfer well from high-resource sources. Overall, DistilGPT-2 and LoRA play a crucial role in mitigating data and hardware scarcity. They enable rapid fine-tuning with mixed-precision training and modest VRAM usage, with only a limited loss in downstream quality [30].

Measuring vocabulary richness is essential in low-resource languages. This helps to ensure that corpora adequately capture lexical diversity for reliable linguistic analysis and model development [31]. Several indices have been proposed to assess vocabulary richness, each offering distinct perspectives. The most basic measure is the type–token ratio (TTR), which reflects the proportion of unique words relative to total tokens. While useful, TTR declines as corpus size increases, limiting its reliability for large datasets [31]. A related measure, hapax

legomena, which counts words that occur only once, highlights lexical creativity and emphasizes the contribution of rare vocabulary words [20], [32]. To overcome the influence of text length, adjusted indices are often employed. Herdan's C, calculated as log(V)/log(N), remains relatively stable across different corpus sizes [31]. Similarly, Guiraud's R ($V/\sqrt{N}$) compensates for size effects and enables meaningful comparisons across datasets [33]. Yule's K measures the concentration of word usage, where lower values indicate even lexical distribution and higher values reflect repetition [34]. Maas's a^2 refines TTR by removing its dependence on text length, with smaller values corresponding to richer vocabularies [35]. Finally, Honoré's H incorporates hapax legomena, making it especially sensitive to rare vocabulary and valuable for examining stylistic variation [32]. Taken together, these indices provide complementary insights into lexical richness of a corpus. By combining them, researchers can achieve a more accurate and nuanced analysis.

In LLM development, a scale of about 100M parameters has become a pragmatic baseline, largely because the widely adopted checkpoints near this size have shown strong fluency and generalization ability at low cost [14], [36]. Research suites deliberately include 70M to 160M parameter variants to enable controlled ablations and reproducible baselines [37]. For generative work, the open models in the 100 to 125M parameters range offer practical starting points [38].

The most common evaluation metrics for question-answering models are perplexity, the Bilingual Evaluation Understudy (BLEU) score, and the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score. Perplexity measures how well a language model predicts a sequence of tokens. It is defined as the exponentiated average negative log-likelihood over the sequence length; lower perplexity indicates better model performance [39]. It is given by:

$$PPL(X) = exp\ \{-\frac{1}{t}\sum_i^t log\ p_\theta(x_i/x_{<i})\} \quad (1)$$

BLEU evaluates the machine-translation outputs using the geometric mean of modified n-gram precisions with a brevity penalty to discourage overly short outputs. A higher BLEU score indicates a closer correspondence to the reference translation [40]. It is given by:

$$BLEU = BP * \exp\left(\sum_{i=1}^{N} (w_i * ln\ (p_i)\right) \quad (2)$$

ROUGE is a family of lexical-overlap metrics that compare a candidate text to one or more reference texts; common variants include ROUGE-N (n-gram overlap), ROUGE-L (longest common subsequence), and ROUGE-S or SU (skip-bigrams, with SU adding unigrams). The results of ROUGE are typically reported in the form of recall, precision, and F1 score [41], [42].

This paper focuses on building a small-scale, transformer-based LLM for Sepedi text generation and question answering. Priority is given to curating sufficient training data to establish a strong baseline model, and its performance is assessed using the relevant evaluation metrics.

III. METHODOLOGY

A. Dataset Selection

The compilation of the required Sepedi text dataset for training the proposed large language model development was drawn from multiple sources, including websites, Hugging Face, GitHub, and internet-scraped text containing Sepedi language content. Some of the datasets include the South African News Data (AfricaNLP), published by Marivate and Sefara [43], which provides Sepedi and Setswana news texts from SABC. Another dataset is the Sepedi NER Corpus, developed by CTexT at North-West University [44], which contains Sepedi government texts annotated for named entity recognition using the CoNLL standard. Additionally, the godlytalias dataset offers a cross-lingual knowledge base and question-answering resource for four low-resource South African indigenous languages, including the Sepedi language [45]. Significant units of text additions were drawn from the National Centre for Human Language Technologies for Northern Sotho (NCHLT.NSO) repository and 390 Sepedi/Northern Sotho past examination papers, from which the text was extracted using a text-extraction script. The integration logs confirm successful de-duplication and format normalization across all the text sources used. In addition, a catalogue of 39 further Sepedi-related resources (including online PDF books and study guides) was compiled.

B. Data Preprocessing

Text preprocessing and dataset normalization were performed in Google Colab using Python, Natural Language Toolkit (NLTK), and regular expressions for data pre-processing. The collected data sources were in heterogeneous structures, encodings, and formats such as Parquet in UTF-8/UTF-16, JSON, TXT, Excel, Word, and PDF. Each dataset was ingested and cleaned independently, then merged into a single UTF-8 CSV corpus after standardizing encodings to UTF-8. Regex-based cleaning was used to remove special characters (@, #, $, %), digits (0–9), Unicode symbols (©, ®, ±), punctuation (“ ”, ‘ ’), HTML tags, emojis, hashtags, alphanumeric clutter, and excess whitespace. The linguistically relevant diacritics (š or Š) found in the Sepedi language were explicitly preserved when detected, while duplicate records were identified and removed. To handle English/Sepedi mixing, an English lexicon (Eng_dictionary.csv with 370,098 words) was loaded and tokenized with the regex pattern [A-Za-z']+; rows with ≥0.80 dictionary coverage and at least two tokens were labelled as English and excluded.

Normalization was used to convert all text to lowercase. Stop words were retained, as they are crucial for grammar and context in text generation tasks. Studies show that removing them degrades the performance in deep learning models like BERT and LSTM [46]. Jurafsky and Martin [39] noted that function words mark grammatical relations, which is essential for fluent text generation. Experts also stress that eliminating stop words reduces the text fluency and coherence [47]. The scaling laws for neural language models showed that the performance of this model depends directly on the data quantity and quality [48], [49], thereby making the arbitrary word removal harmful. Tokenisation was carried out using tools such as *Hugging Face Tokenizers*, *SpaCy*, and the NLTK. The process has applied techniques including the word-based, sentence-based, and phrase-based tokenisation methods. Language modelling techniques like the *Byte-Pair Encoding* (BPE) and the *SentencePiece* were utilised to efficiently segment the text into meaningful units, preparing the text data for effective large language model training.

C. Exploratory Data Analysis (EDA)

The following results were recorded from the data preprocessing. The cleaned corpus was analysed in Python using *matplotlib* and *seaborn* for charts, *wordcloud* for word clouds, and the NLTK (with the *punkt* tokenizer) for n-gram utilities**.** The word cloud in Fig. 2 presents a sample of 4-7 letter tokens that were extracted from the dataset. The results show that the dataset is constituted of frequent and meaningful Sepedi word tokens, which include both function words (e.g., *gore*, *gomme*) and content words (e.g., *afrika*, *karolo*). This balance underscores the corpus's linguistic authenticity, thereby reflecting an everyday language use and cultural context. The variation across visualizations has further confirmed the correctness of the tokenization step, as the most frequent tokens are valid Sepedi words rather than symbols, digits, or mixed-language noise. The prominence of 4–7 letter tokens also illustrate the morphological richness and agglutinative nature of the Sepedi language, thereby ensuring that both the structural and semantic patterns are well represented.



Fig. 2 The word cloud of tokens consisting of 4, 5, 6 and 7 letters.

Fig. 3 presents a bar graph of the top ten most frequent words in the Sepedi training dataset.

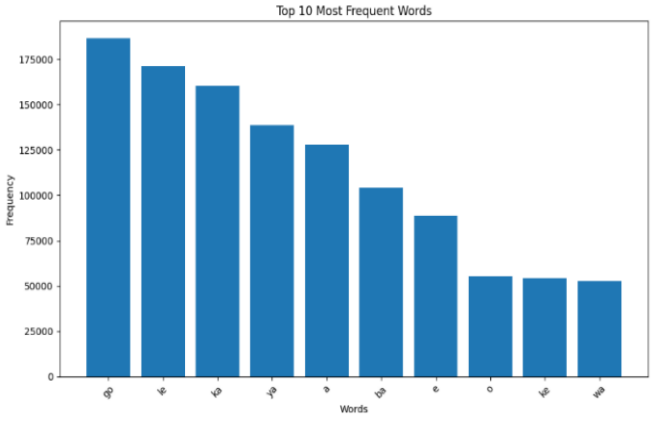


Fig. 3. Top 10-word frequencies in the training dataset.

The y-axis shows the frequency of occurrence, and the x-axis lists the words ranked from highest to lowest. The distribution shows a sharp decline after the most common tokens, thereby reflecting the natural Zipfian trend in language data [50]. It is evident that stop words dominate at the top ranks, which indicates their central role in the structure and flow of Sepedi text. This suggests that the dataset has effectively captured the recurrent Sepedi language patterns, which will be beneficial for the effective training of the large language model that learns both syntactic and semantic dependencies.

The distribution of token counts in Fig. 4 illustrates that there is a descending trend in token counts, where the lower-order n-grams (2–4) dominate in frequency while the higher-order n-grams (5–10) gradually decrease. This is consistent with the observations in computational linguistics that shorter n-grams occur more frequently due to their role in capturing function words and common collocations [51], [52]. Although less frequent, higher-order n-grams provide richer semantic and contextual information that supports tasks such as language modelling and morphological analysis [43], [54]. The balanced decline across the n-gram sizes suggests that the dataset preserves both high-frequency lexical patterns and longer phrasal structures, which is essential in developing a robust large language model training process [55].

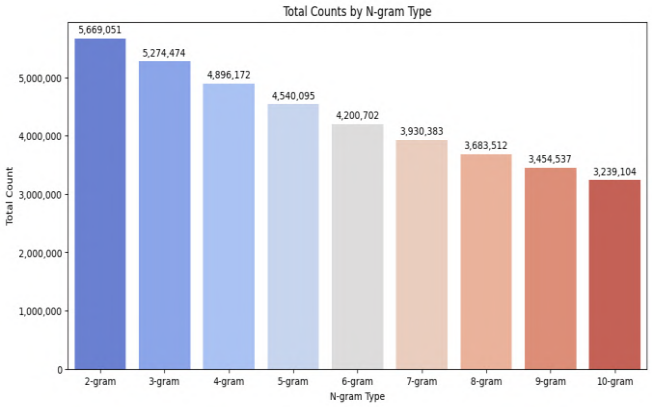


Fig. 4. The graph shows the total n-gram count across different n-gram types.

The n-gram heat maps were used as an additional data-analytic method, to investigate the linguistic behaviour that is embedded in the training corpus. The final step of the data analysis was to compute the vocabulary richness of the corpus, as shown in TABLE I.

TABLE I: VOCABULARY RICHNESS

Metrics	***Values***
Characters	33 548 934.00
Tokens (N)	6 356 302.00
Types (V) TTR	134 612.00 0.021
Hapax V1	56 753.00 (rate: 0.008929)
Herdan C	0.753
Guiraud R	53.39
Yule K	133.00
Maas a^2	0.012
Honoré H	2,708.35

After performing a corpus-level vocabulary richness data analysis, the dataset was found to contain 33.5 million characters and approximately 6.36 million tokens, with 134,612 unique types. This analysis indicates that the dataset provides a strong foundation for LLM training. The low type-token ratio (0.021) is expected given the corpus size and does not imply lexical limitation; instead, the stability of length-

adjusted indices (Herdan's C = 0.754, Guiraud's R = 53.39) suggested that the dataset sustains a relatively high degree of lexical diversity even at scale. The hapax rate (0.0089) and Honoré's H (2708.35) confirm that rare words are sufficiently represented, contributing to vocabulary breadth without overwhelming the model with noise. Measures of repetition (Yule's K = 133.0, Maas's a^2 = 0.0157) indicate a balanced distribution between frequent and infrequent terms, which is advantageous for generative training: the model can capture core linguistic patterns while still being exposed to less common lexical items. This balance helps reduce overfitting to repetitive tokens and supports richer, more contextually adaptive text generation. In overall, the results highlight that the dataset combines a broad coverage with structured repetition, characteristics that are well aligned with the requirements of large language model training for robust and varied text generation [56].

D. Training the Model

The proposed Sepedi text-generation LLM (question answering) was trained using the Google Colab environment for small-scale training. The LLM was fine-tuned from a small GPT-2 (DistilGPT-2) using LoRA for parameter-efficient adaptation. The training process was conducted by using a single NVIDIA T4 GPU, with an FP16 and gradient checkpointing. The Hugging Face Trainer ran AdamW (base LR 2×10^{-4}) with cosine decay and a 5% warmup. Because of VRAM limits, batching used a per-device batch size of 2 with gradient accumulation of 8 (effective when it is at 16) at a sequence length of 512. The training runs covered 10, 50, 100, 150, and 300 epochs, with periodic checkpoints to support the Colab resumability in case the training process was interrupted.

The dataset was split 80/10/10 (train/validation/test). A portion of the dataset was structured into a Q&A format ("Potšiso-karabo"), featuring one-to-many question–answer mappings. In addition, the short context passages were placed between each question and its answers to improve clarity. The following example illustrates the record format:

```
[
  {
    "q": "question",
    "context": "sentences or words or phrases related to the question"
    "a": "answer_1\n answer_2\n answer_3\n ...answer_n"
  },
    # followed by another list of Q and A #
]
```

The remainder of the corpus (6M words) was used for language modelling to better capture the lexical patterns of a fluent Sepedi language. The 82.7M-parameter model was tested and evaluated using the perplexity, BLEU, and ROUGE evaluation metrics.

IV. Preliminary Results.

A. The Epoch Experiment

The epoch experiment spanned 10, 50, 100, 150, and 300 epochs while varying batch size and learning rate. The training and validation losses declined together, dropping rapidly from about 40 to around 10 in the first 100 epochs and reaching roughly 5 by the 120–150 epoch range, as illustrated in Fig. 7. After this point the curves flattened, approaching near-zero between about 230 and 260 epochs, while the gap between training and validation stayed small, thereby indicating a stable generalization with no visible overfitting. Taken together, this implies that the Sepedi text-generating Q&A model has internalized generalizable patterns and should produce coherent, context-aware answers with extended training beyond the 150-epoch size.

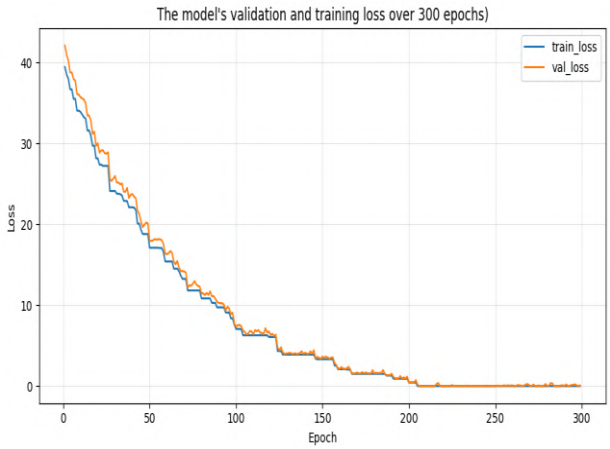


Fig. 7. The results of training and validation loss for 300 epochs.

B. Evaluation Metrics

The performance of the developed Sepedi text-generation LLM was evaluated using the standard metrics: perplexity, ROUGE, and BLEU. The first metric is perplexity, which measures how well the model can predict the next token in a sequence, with the lower values indicating better predictive performance.

The graph in Fig. 8 presents the model's perplexity, which indicates that the Sepedi Q&A LLM converges well; training and evaluation perplexity both decline and remain closely aligned, therefore implying minimal overfitting despite early zigzags from learning-rate scheduling and batch variance. The model's performance improves steadily, and the best validation perplexity, which is indicated by the dotted line at the 291 epochs, marks the checkpoint at which the model is saved and prepared for deployment.

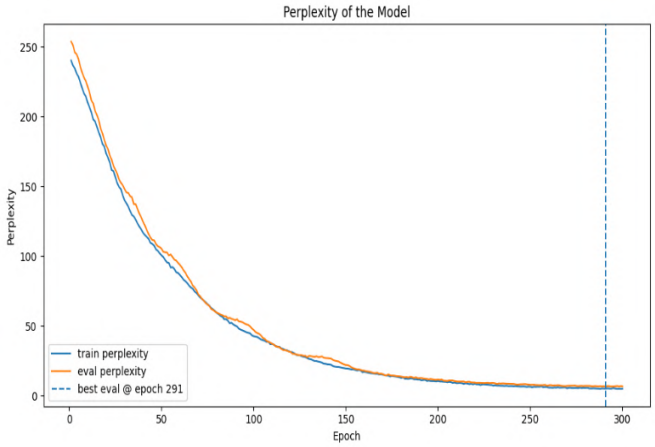


Fig. 8. The results for model's perplexity over 300 epochs.

The second metric is BLEU, which checks how many short word sequences in the model's text match the reference and penalizes very short outputs; higher scores mean better precision. The third metric is ROUGE, which measures how much of the reference the model covers (for example, ROUGE 1 and ROUGE L), whereby the higher scores mean better recall. The graph in Fig. 9 shows that BLEU is about 0.40, ROUGE 1 about 0.21, and ROUGE L about 0.19. The curves dip early around steps 0 to 40 epochs, then rise, and subsequently speed up near 110 to 180 epochs, and then level off after about 220 epochs, with the remaining zigzags matching batch or optimizer noise rather than real gains. The steady gap where BLEU is higher than ROUGE L suggests strong local precision but weaker long-range recall and coverage, so the outputs are accurate in small pieces yet may leave out parts of multi-part answers responsible for addressing several related sub-questions in one response.

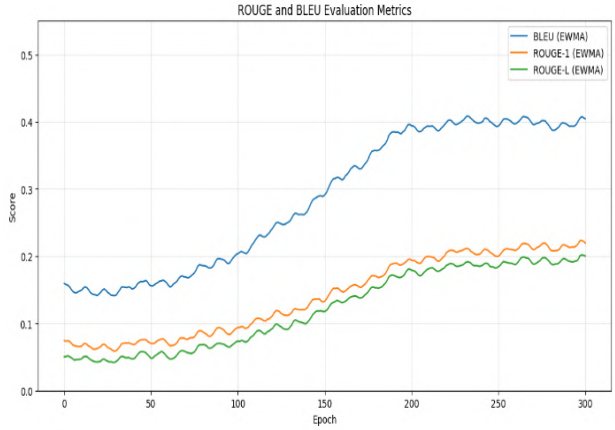


Fig. 9 A line graph showing the ROUGE and BLEU evaluation metrics.

C. The Analysis

The model training experiment provided clear insights into the behaviour of the Sepedi text-generation model under varying data and training conditions. The model has 82.7 million parameters, which is not far from the benchmark target for an LLM as outlined in the literature. The corpus diagnostics showed a strong lexical authenticity, with everyday Sepedi tokens predominant, and length-adjusted richness indices (Herdan's C and Guiraud's R) indicating a balanced textual diversity. The Zipfian and n-gram profiles confirmed well captured recurrent patterns in the dataset. For model training across the 10 to 300 epochs, the training and validation losses fell sharply until they reached around 150 epochs, and then they plateaued, with a small and stable gap that indicated an effective learning process of the model without overfitting. Increasing the size and variety of the dataset exposed the model to broader vocabulary and writing styles, which reduced overfitting and made the validation loss curve smoother and closer to the training curve. The preservation of the diacritics and function words helped in retaining the key grammatical and phonological cues. The LoRA fine-tuning has enabled a stable optimisation within the current small computational constraints, as evidenced by fewer loss spikes and steadier convergence.

The model's evaluation results aligned with these trends. Perplexity declined in tandem on the training and validation splits, and on a held-out question and answer subset the model achieved BLEU 0.40), ROUGE 1 (0.21), and ROUGE L (0.19). BLEU was notably higher than ROUGE L, signalling a strong local n-gram precision but weaker long-range coverage, which yielded fluent yet sometimes incomplete answers that required multiple facts.

The current metrics remain unsatisfactory but are expected to improve as the project progresses. Suboptimal results from this small-scale experiment stem from the compute constraints in Google Colab, which necessitated the use of a compressed model and may have degraded training quality. Although this is a lightweight, low-cost setup that is suited for the initial feasibility testing, it is not ideal for the long-context question-answering text generation task, where a stronger and more consistently available compute is required. This limitation is a temporary constraint that prevented attainment of optimal results. To improve the model's performance, the model will be scaled from 82.7M to 100M parameters while improving the data size through the ongoing data collection and augmentation. Targeted data augmentation will include the token-level methods (word substitution, token dropout), sentence-level methods (paraphrasing, back-translation), and adversarial variants (mixup, interpolation), with effects evaluated on longer-context benchmarks. In addition, curriculum scheduling and domain-adaptive pretraining (DAPT) are expected to improve discourse-level coherence and narrow performance gaps on long-form tasks. All experiments will run on a high-performance computing cluster for a full-scale model of the Sepedi text-generating question-answering LLM.

V. CONCLUSION

This research work has developed a transformer-based Sepedi text-generating LLM for question-answering. This was achieved by addressing data scarcity, uneven digital representation, and the limited computational infrastructure dedicated to the under-resourced languages. A large multi-sourced Sepedi corpus was collected, curated and pre-processed to train this LLM. The GPT-2 (DistilGPT-2) transformer was used to train a baseline question-answering, text-generating LLM with a parameter-efficient fine-tuning process. The model was evaluated using perplexity, BLEU, and ROUGE evaluation metrics. These metrics showed a moderate model performance whereby optimal results were hindered by compute constraints on Google Colab, which affected the training process. These constraints constitute the principal limitation of the current work. Upcoming work will expand the dataset through additional collection and targeted data augmentation, followed by training the Sepedi question-answering LLM on a high-performance computing cluster for full-scale development and deployment. Building on these findings, the future work in this field should focus on broadening the corpus coverage across various domains and registers, which includes varied writing styles and levels, from academic to informal. Secondly, there must be a focus on developing a culturally informed evaluation of the model.

ACKNOWLEDGEMENTS

This study was supported by the Department of Computer Science at the University of Limpopo, in collaboration with the Telkom Centre of Excellence for Speech Technology and the Telkom–National Research Foundation (Telkom-NRF) Future Technologies Programme.

REFERENCES

[1] A. Mehrish, N. Majumder, R. Bhardwaj, R. Mihalcea, and S. Poria, "A review of deep learning techniques for speech processing," 2023.

[2] Koushiki, "Building language models in NLP," Analytics Vidhya, 2022.

[3] C. Wang, J. Zhao, and J. Gong, "A survey on large language models from concept to implementation," 2024.

[4] J. Li, T. Tang, W. X. Zhao, and J.-R. Wen, "Pretrained language models for text generation: A survey," *arXiv preprint* arXiv:2105.10311, 2021.

[5] L. Yu, L. Sartran, W. Stokowiec, W. Ling, L. Kong, P. Blunsom, **et al.**, "Better document-level machine translation with Bayes' rule," Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL), pp. 5017–5033, 2020.

[6] L. Huang, L. Wu, and L. Wang, "Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward," Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL), pp. 5094–5107, 2020.

[7] T. Iqbal and S. Qureshi, "The survey: Text generation models in deep learning," J. King Saud Univ.—Comput. & Inf. Sci., vol. 34, pp. 1046–1062, 2022.

[8] M. Wong, "The AI revolution is crushing thousands of languages," The Atlantic, 2024.

[9] I. Adebara, A. Elmadany, and M. Abdul-Mageed, "Cheetah: Natural language generation for 517 African languages," 2024.

[10] J. O. Alabi, D. I. Adelani, M. Mosbach, and D. Klakow, "Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning," arXiv preprint arXiv:2204.06487, 2022.

[11] M. A. Masethe, H. D. Masethe, S. O. Ojo, and P. A. Owolawi, "Paraphrase generation model using transformer-based architecture," SSRN Electron. J., 2024.

[12] S. P. Ramalepe, T. I. Modipa, and M. H. Davel, "Pre-training a transformer-based generative model using a small Sepedi dataset," 2025.

[13] J. McGiff and N. S. Nikolov, "Overcoming data scarcity in generative language modelling for low-resource languages: A systematic review," 2025.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," Proc. NAACL-HLT 2019, pp. 4171–4186.

[15] W. Nekoto, V. Marivate, T. Matsila, J. F. Abbott, D. I. Adelani, J. M. Akinola, et al., "Participatory research for low-resourced machine translation: A case study in African languages," arXiv preprint arXiv:2010.02353, 2020.

[16] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, et al., "Unsupervised cross-lingual representation learning at scale," Proc. ACL 2020, pp. 8440–8451.

[17] K. Siminyu, G. Kalipe, D. Orlic, J. Abbott, V. Marivate, S. Freshia, et al., "AI4D -African language program," 2021.

[18] D. I. Adelani, J. Abbott, G. Neubig, D. D'souza, J. Kreutzer, C. Lignos, et al., "MasakhaNER: Named entity recognition for African languages," Trans. Assoc. Comput. Linguistics, vol. 9, pp. 1116–1131, 2021.

[19] T. Ehsan and T. Solorio, "Enhancing NER performance in low-resource Pakistani languages using cross-lingual data augmentation," 2025.

[20] J. Chen, D. Tam, C. Raffel, M. Bansal, and D. Yang, "An empirical survey of data augmentation for limited data learning in NLP," 2021.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., "Attention is all you need," Adv. Neural Inf. Process. Syst., vol. 30, pp. 5998–6008, 2017.

[22] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, et al., "LLaMA: Open and efficient foundation language models," arXiv preprint arXiv:2302.13971, 2023.

[23] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, et al., "Multilingual denoising pre-training for neural machine translation," Trans. Assoc. Comput. Linguistics, vol. 8, pp. 726–742, 2020.

[24] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers in natural language processing," *Artificial Intelligence Review*, vol. 54, no. 1, pp. 1–47, 2021.

[25] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, et al., "Language models are few-shot learners," Adv. Neural Inf. Process. Syst., vol. 33, pp. 1877–1901, 2020.

[26] M. A. Hedderich, D. I. Adelani, D. Zhu, J. Alabi, U. Markus, and D. Klakow, "Transfer learning and distant supervision for multilingual transformer models: A study on African languages," Proc. EMNLP 2020, pp. 2580–2591.

[27] S. Mesham, L. Hayward, J. Shapiro, and J. Buys, "Low-resource language modelling of South African languages," 2021.

[28] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient finetuning of quantized LLMs," arXiv preprint arXiv:2305.14314, 2023.

[29] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," arXiv preprint arXiv:2106.09685, 2021.

[30] L. Tunstall, L. von Werra, and T. Wolf, *Natural Language Processing with Transformers*, Revised ed., O'Reilly Media, 2022

[31] K. Kettunen, "Measuring lexical diversity in large corpora: A computational study," Proc. Nordic Conf. Computational Linguistics (NoDaLiDa), 2023, pp. 481–493.

[32] Y. Chen, L. Wang, and H. Xu, "Lexical richness and language proficiency: Revisiting traditional indices with modern corpora," J. Appl. Linguistics Lang. Res., vol. 10, no. 2, pp. 55–73, 2023.

[33] P. McCarthy, "Exploring Guiraud's index as a robust measure of lexical diversity," UWE Research Repository, 2021.

[34] A. Verma, "Revisiting Yule's K: Applications in computational stylistics," Comput. Linguistics, vol. 48, no. 3, pp. 601–620, 2022.

[35] J. Zhou and F. Li, "Evaluating Maas's a^2 and Honoré's H in big data text analysis," J. Machine-Assisted Intelligence, vol. 5, no. 1, pp. 22–39, 2024.

[36] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J. (2020) 'Exploring the limits of transfer learning with a unified text-to-text transformer', *Journal of Machine Learning Research*, 21(140), pp. 1–67.

[37] Biderman, S., Schoelkopf, H., Anthony, Q., Bradley, H., O'Brien, K., Hallahan, E., Khan, M.A., Purohit, S., Sai Prashanth, U., Raff, E., Skowron, A., Sutawika, L. and van der Wal, O. (2023) 'Pythia: A suite for analyzing large language models across training and scaling', *arXiv preprint* arXiv:2304.01373.

[38] Black, S., Gao, L., Wang, P. and Biderman, S. (2021) 'GPT-Neo: Large-scale autoregressive language modeling with Mesh-TensorFlow', Technical report, EleutherAI.

[39] D. Jurafsky and J. H. Martin, Speech and language processing, 3rd ed. draft, Stanford Univ., 2021. [Online].Available: https://web.stanford.edu/~jurafsky/slp3/

[40] Lee, S., Lee, J., Moon, H., Park, C., Seo, J., Eo, S., Koo, S. & Lim, H., 2023. A survey on evaluation metrics for machine translation. *Mathematics*, 11(4), 1006. https://doi.org/10.3390/math11041006

[41] Bhandari, M., Gour, P., Ashfaq, A., Liu, P. & Neubig, G., 2020. Re-evaluating evaluation in text summarization. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics. Available at: https://arxiv.org/abs/2010.07100 [Accessed 12 Sep. 2025].

[42] Fabbri, A.R., Kryściński, W., McCann, B., Xiong, C., Socher, R. & Radev, D., 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9, pp. 391–409. https://doi.org/10.1162/tacl_a_00373

[43] V. Marivate and T. Sefara, "South African news data," Zenodo, 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3668495.

[44] Community Datasets, "sepedi_ner," Huggingface, 2016. [Online]. Available:https://huggingface.co/datasets/community-datasets/sepedi_ner

[45] S. Ralethe, "Knowledge_Base_Projection," Huggingface, 2025. [Online].Available:https://huggingface.co/datasets/sello-ralethe/Knowledge_Base_Projectin

[46] A. Miyajiwala, A. Ladkat, S. Jagadale, and R. Joshi, "On sensitivity of deep learning based text classification algorithms to practical input perturbations," arXiv preprint arXiv:2201.00318, 2022.

[47] J. Howard, "Discussion on stop words in NLP models," fast.ai forum, 2020. [Online]. Available: https://forums.fast.ai

[48] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?," in Proc. ACM Conf. Fairness, Accountability, and Transparency, 2021, pp. 610–623.

[49] J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, F. Song, et al., “Scaling language models: Methods, analysis & insights from training Gopher,” arXiv preprint arXiv:2112.11446, 2021.

[50] Y. He, Q. Zeng, and M. Jiang, “Pre-trained models perform the best when token distributions follow Zipf’s law,” arXiv preprint arXiv:2507.22543, 2025.

[51] E. M. Bender and A. Koller, “Climbing towards NLU: On meaning, form, and understanding in the age of data,” Trans. Assoc. Comput. Linguistics, vol. 8, pp. 556–572, 2020.

[52] C.-H. Chiang, Y.-T. Yeh, and H.-H. Chen, “Pre-training with phrase and n-gram masking for Chinese BERT,” IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 30, pp. 3173–3184, 2022.

[53] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, “Efficient transformers: A survey,” ACM Comput. Surveys, vol. 55, no. 6, pp. 1–44, 2022.

[54] L. Ranaldi, D. Brunato, and F. Dell’Orletta, “Exploring multiword expressions in pre-trained language models: A case for n-gram representations,” in Proc. EACL 2023, pp. 3129–3140.

[55] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing,” ACM Comput. Surveys, vol. 55, no. 9, pp. 1–36, 2023.

[56] K. Lee, D. Ippolito, A. Nystrom, C. Zhang, D. Eck, C. Callison-Burch, et al, “Deduplicating training data makes language models better,” *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, pp. 8424–8445, 2022.

Mathews Tladi is pursuing a postgraduate degree in Master of Science in Computer Science at the University of Limpopo, focusing on large language models (LLMs). His research interests include Natural Language Processing (NLP), Deep Learning and Machine Learning and their applications within Artificial Intelligence.

Madimetja Jonas Manamela is a senior lecturer in the Department of Computer Science at the University of Limpopo (UL). His research publications focus on Natural Language Processing (NLP) and Speech and Language Processing (SLP) within Artificial Intelligence.

Thipe Modipa is a senior lecturer in the Department of Computer Science at the University of Limpopo (UL). He received his PhD (Information Technology) from the North-West University. His research interests include code-switching for under resourced languages, text generation, automatic speech recognition.

A Cross-device Horizontal Federated Learning with an asynchronous-based aggregation mechanism in detecting Secure Shell Brute Force attacks for Internet of Things in Edge Computing

Miyelani Silence Madiba
Department of Computer Science
University of Limpopo
Polokwane, South Africa
silence.madiba@ul.ac.za

Mthulisi Velempini
Department of Computer Science
University of Limpopo
Polokwane, South Africa
mthulisi.velempini@ul.ac.za

***Abstract*—Cloud computing is a promising technology. However, the Cloud is currently facing growing challenges in handling the data generated by the IoT's and other associated applications. The Cloud is attempting to ensure low inflexibility and efficient data transfer rate utilisation, optimal use of computational resources, and flexible and essential functionality for IoT gadgets while moving the entire data to the cloud, given that billions of already disconnected devices are currently producing several exabytes in data daily. It becomes essential to protect user data. This study tackles the problem of identifying Secure Shell (SSH) Brute Force attacks offering a novel way to improve cybersecurity in Internet of Things (IoT) contexts. We provide a framework called Cross-device Horizontal Federated Learning (CHFL) that allows network-edge IoT devices to work together to train a global model without disclosing their private, local information. Our framework's asynchronous-based aggregation technique, which enables devices to publish their model changes independently of one another, greatly increases the effectiveness and scalability of model training. Our experimental results, validated using the CSE-CIC-IDS2018 dataset, show that this method outperforms LSTM and FAil2ban algorithms in terms of accuracy and other metrics.**

Keywords—cloud computing, edge computing, internet of things, brute force attack, machine learning

I. INTRODUCTION

In recent years, the concept of Cloud Computing (CC) has emerged as one of the key paradigms. CC has emerged as an important advancement with significant implications for the whole information technology (IT) industry, not simply internet services. [1]. However, The present cloud model finds it challenging to adapt to new demands due to the growing popularity of Internet-of-things (IoT) technologies and the high-performance computing requirements of contemporary end-user applications. The cloud concept provides strong network connectivity and users can access remote data centres, including storage and computation resources, to cut expenses on planning, acquiring, and maintaining these resources [2] [3]. However, the Cloud is currently facing growing challenges in handling the data generated by the IoT's and other associated applications. The Cloud is attempting to ensure low inflexibility and efficient data transfer rate utilisation, optimal use of computational resources, and flexible and essential functionality for IoT gadgets while moving the entire data to the cloud, given that billions of already disconnected devices are currently producing several exabytes in data daily. It becomes essential to protect user data [4].

With the challenges the CC faces, an emerging computing paradigm called Edge Computing (EC) has been presented as an important enhancement to CC. Its purpose is to increase the computing and storage capacity at the edge of the IoT. IoT devices can particularly obtain resources in EC by delegating work to adjacent edge servers. Edge servers are located closer to IoT devices that is, closer to the data sources than cloud servers. This mitigates, to some degree, the privacy leakage risk associated with communication over long distances [5] [6]. Furthermore, the computational, storage, and bandwidth constraints of conventional CC can be lessened by efficiently integrating and using the resources of edge devices. Thus, in smart cities, smart grid, smart healthcare, and other IoT situations, EC can effectively provide distributed and low-latency computing capabilities [7] [8].

Secure Shell (SSH) offers an efficient login method for remote users. Brute-force attacks, in which an attacker repeatedly attempts a large number of passwords, can be used to compromise the password-based authentication that these users utilised. These attacks may come from just one source or from several sources cooperating [9]. Gathering usernames is an essential starting point in any attempt to enter or initiate an attack on a victim system or service. Thus, a perpetrator initially collects the essential details about a user. Targeted password-based attacks can be performed against discovered usernames after the planned usernames have been listed [10]. Therefore there id a need for efficient SSH brute force attacks detection scheme.

Fig. 1. Presents a simplified model of an SSH brute-force attack, demonstrating how an intruder can use a network of compromised computers to obtain unauthorised access to servers and Internet of Things (IoT) devices. The process starts with the "Attacker Infrastructure," where a hacker oversees a botnet, a group of compromised computers that serve as attack agents. These bots are instructed by a central "Botnet C&C Server" to launch a coordinated attack on a target network, specifically targeting devices with open SSH ports. The attack is represented as a series of repeated attempts, where the bots methodically try various username and password combinations until an appropriate match is found.

The diagram's "IoT device" portion, which demonstrates the brute-force approach in action, highlights the attack's

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

evolution. The broken locks, which indicate unsuccessful credential guesses, are the first indication that the attacker has sent several login attempts. The green checkmark and lock icon in the last step signify a successful breach, indicating that the attacker has successfully authenticated and taken over the device. By showing how a single attacker can use a large number of compromised computers to overwhelm a target's defences and eventually take control of their network resources and associated devices, the figure successfully illustrates the scope of such an attack.

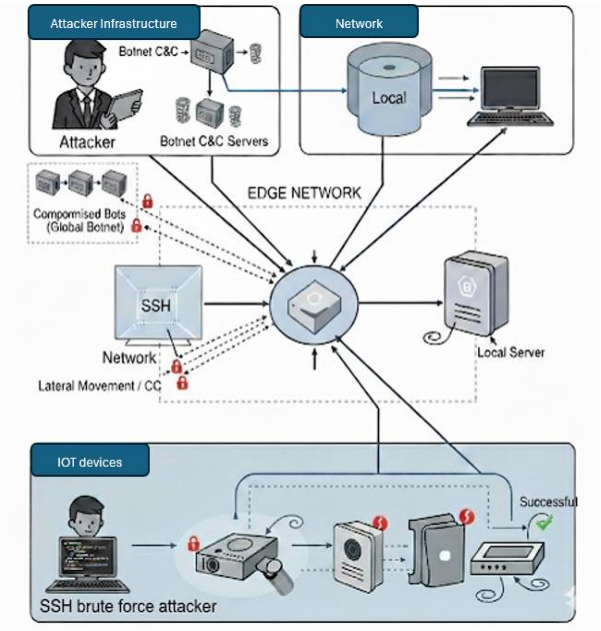


Fig. 1. Accuracy Comparison Results

The aim of this study is to develop a Cross-device Horizontal Federated Learning (CH-Federated Learning) with an asynchronous-based aggregation mechanism in detecting Secure Shell Brute Force attacks for IoT in EC.

The contribution of the paper is as follows:

- The paper contributes to the developing a CH-Federated Learning that will best detect the SSH brute force attacks.
- The paper also compare the developed model to other existing models in the literature using different metrics.

In this section, the study was introduced, starting with CC and how EC came to offload the cloud. We also introduce SSH brute force attack and how it works. In Section 2, we examine relevant literature, studies that have also attempted to solve the problem, detecting the brute force attacks. In Section 3, that where we will discuss our proposed model and its justifications. In section 4, we look at the methodology, the methods we used to perform our simulations. Section 5, we demonstrate the results, where we compared our model with other 2 models found in the literature. Finally in section 6 we conclude the paper.

II. RELATED WORK

Thise section looks at related work, we divided it into 2 sections; traditional approaches for SSH brute force attack

A. Traditional tools SSH brute force attack detection

1) Fail2ban

A security program called Fail2Ban keeps monitoring on system logs for recurring unsuccessful login attempts. It modifies firewall rules to temporarily disable the violating IP address when it notices questionable activity. This program works especially well to defend against brute-force attacks on systems like FTP and SSH [11]. Fail2Ban's flexibility is one of its key advantages; administrators may specify which logs and services to keep an eye on, establish criteria for unsuccessful attempts, and alter the length of bans. Because of its portability and low system resource requirements, it can be used in a variety of server contexts [11] [12].

Fail2Ban is not without its limits, either. It works well against straightforward brute-force threats, yet it is less successful against dispersed attacks which aim at non-standard APIs or use many IP addresses. Additionally, legitimate users who frequently type incorrect passwords run the danger of being inadvertently blocked. Furthermore, since Fail2Ban depends on precise log parsing, fraudulent behaviour might go unnoticed if log standards are altered or incorrectly configured. Finally, since suspensions are usually just short, persistent attackers might just come back when the ban time is up. An independent system tool called a firewall regulates traffic by permitting or prohibiting IP addresses and ports. Instead than replacing the firewall, Fail2Ban complements it by automatically adding block rules based on unsuccessful login attempts [11] [13].

2) PAM Tally2

The Pluggable Authentication Module (PAM) system used on Linux systems includes the PAM Tally2 module. In order to slow down brute-force breaches aimed at legitimate usernames, it keeps records of the number of unsuccessful login attempts for each user account and has the ability to freeze an account after a predetermined number of failures. Enforcing account lockout regulations at the local system level is PAM Tally2's primary benefit. It does not, however, handle distributed or network-level threats; it primarily defends against attacks on known usernames.

Enforcing local account lockout restrictions, which provides an extra degree of security at the authentication level, is one of PAM Tally2's main advantages. It is part of the Linux authentication stack and can be set up to reset counters automatically following successful logins. However, PAM Tally2 is not immune to distributed violations that affect several accounts or attacks aimed at the network layer. Extensive configuration might result in unexpected lockouts, such as the locking out of admins or service accounts. Furthermore, several Linux distributions are replacing the module with more recent versions due to problems with improper tally increments [11] [14].

B. Machine Learning tools SSH brute force attack detection

1) Convolution Neural Network (CNN)

CNN is a kind of neural network framework that operates using the sharing of weight concept. It uses the associated properties to determine a specific weight for every node. CNN is most effective at handling two-dimensional input, including matrix data, video files, and pictures. Because the data is two-dimensional, it may therefore be utilised for detecting SSH attacks. CNN employs a number of extraction

of features strategies with the use of hidden layers. It aids in learning how data is represented. The three stack levels that make up CNN's design are referred to as the convolution layer, the pooling layer, and the entirely linked layer [15].

2) Long Short-Term Memory (LSTM)

One kind of recurrent neural network is called an LSTM. Hochreiter and Hochreiter launched it in 1997. In essence, LSTM operates by a feedback system. It can analyse several input data types, including speech and video data, unlike CNN. Four stacked layers that are next to one another and are provided with a random value are used by LSTM for SSH attacks. An extensive layer containing the SoftMax or sigmoid characteristic coupled is the end result [16].

3) Multi Layer Perceptron (MLP)

Multilayer Perceptron's (MLPs) are ANN models that employ feedforward learning. An MLP is composed of three node layers: the input layer, the hidden layer, and the output layer. Each of the MLP's nodes are neurons, with the notable exception of the first input node. A nonlinear activation process is used by all of these neurons. Because of its numerous layers and nonlinear activation, MLP uses the supervised learning method known as backpropagation when training. MLP does not function similarly to a single layer perceptron. MLP can be used to divide up data inputs that cannot be divided linearly [16] [17].

4) Logistic Regression (LR)

LR is an approach to statistical analysis that predicts a binary result of 0 or 1 based on prior findings from a data collection. In LR the proper output class is determined using a sigmoid function. A LR model predicts a dependent data variable by looking at the association amongst one or more different independent variables that are currently existent [18].

5) Naïve Bayes

The stochastic machine-learning algorithm, Naive Bayes is often utilised in prediction challenges. It operates by determining the likelihood of identifying a particular class in a given dataset, based on the Bayes theorem. Among its elements are class and conditional likelihoods. The ratio of each occurrence of a class to all other instances is known as the class possibility. The conditional likelihood of a class is calculated by dividing its sample occurrence by each feature occurrence. The Naive Bayes classifier considers the relationships between these variables and assumes that each attribute is asymmetric. Although the framework is simple and fast to implement, its major flaw is that it requires the predictors to be independent. In most real-world situations, the predictors rely on one another [17].

6) Random Forest (RF)

RF is another commonly used machine-learning algorithm that falls under the category of supervised learning techniques. Similar to this, RF are used to solve classification problems in machine learning. The classifier's procedure consists of two asymmetric phases. In the initial stage, the given dataset is used to build the asymmetrical forest, and in the second stage, the classifier that was constructed is used to make the prediction. To reduce the degree of association among the various models, decision trees were utilised to train them in a randomised fashion. RF takes low training time, predicts outcome with high accuracy, and offers a reliable feature significance evaluation. On the other hand, training many deep trees can be computationally costly, and RF is inherently less useful than a single DT [17].

7) K -Nearest Neighbour (K-NN)

The K-NN approach relies on the nearest training instances in feature space. A training example and an unknown example are compared using the K-NN approach. The training dataset's characteristic of closeness, which is based on distance formulas in an n-dimensional space, defines it. Second, a majority vote among the neighbours they encounter determines K-NN's ranking. The distance between each neighbour is determined using the most widely used Euclidean distances. K-NN suggests that the final class forecast is influenced by closest neighbours and that similar examples exist side by side. It is an easy technique to put into practice as quickly as possible [19].

III. PRPOSED MODEL

This work proposed a Cross-device Horizontal Federated Learning (VH-federated Learning) with an asynchronous-based aggregation mechanism in detecting Secure Shell Brute Force attacks.

A Cross-device Horizontal Federated Learning (FL) model with asynchronous-based aggregation for detecting Secure Shell (SSH) brute-force attacks is a decentralized machine learning system. It trains a shared model to identify malicious login attempts across numerous devices without collecting their sensitive, local data.

This model uses the following components to function:

A. Cross-device

The model operates across a large number of individual devices, such as laptops, servers, and Internet of Things (IoT) devices, that are running an SSH service. These devices act as the "clients" in the federated learning network.

B. Horizontal Federated Learning (HFL)

This is the FL approach used when participating devices have datasets with the same features but different data samples. In the context of SSH, all devices would have similar login attempt data (e.g., failed attempts, timestamps, source IP addresses) but for different, unique attack instances. The devices collaboratively train a shared model without sharing their raw data.

C. Asynchronous-based Aggregation

This is the key mechanism for combining model updates. Unlike synchronous methods, where the central server waits for all selected clients to finish their training and send updates, an asynchronous system updates the global model as soon as a client's update is received. This is crucial for cross-device FL, where devices may have varying network speeds or computational power, leading to "stragglers" (slow devices). The server doesn't wait for the slow devices; it incorporates the updates from the fast ones immediately.

The process follow the following procedure:

-A central server sends an initial model to a group of devices.

-Each device trains the model locally using its own SSH log data.

-As soon as a device finishes training, it sends its model updates (not the raw data) back to the server.

-The server immediately incorporates the received updates into the global model, making it available for the next devices to download and train on.
-This continuous cycle of local training and immediate global aggregation allows the model to constantly improve its ability to detect brute-force attacks.

The model is Important because
This model is important because it solves two major problems in cybersecurity and machine learning: data privacy and scalability.

Privacy-Preserving Threat Intelligence: SSH brute-force attacks are a common threat. Organizations can't easily share their attack data due to privacy concerns and data regulations. This model allows them to collaborate on a shared security model without ever exposing their sensitive log data. This means a company can benefit from the attack data of hundreds of other companies, collectively improving their defense, while their specific data remains private.

Decentralized Intelligence: Rather than centralizing all log data in one place, which creates a single point of failure and a massive target for attackers, this model distributes the intelligence. The attack detection capabilities reside on each device, making the system more resilient and able to detect attacks even if the central server is down.

Advantages
The use of this specific model for SSH brute-force detection offers several advantages:

a) Enhanced Privacy and Security:

The most significant benefit is that sensitive log data never leaves the device. The model learns from the collective behavior of attacks without needing to see the individual data points (e.g., failed usernames or passwords).

b) Robustness to Stragglers

The asynchronous aggregation mechanism makes the system efficient and robust. It prevents slower devices from bottlenecking the entire training process, which is a common issue in large-scale, cross-device scenarios where device connectivity and hardware capabilities vary widely. The model can continuously learn and improve without waiting.

c) Scalability

The model can easily scale to thousands or even millions of devices. Each device contributes a small update, and the system can handle updates as they arrive, making it highly suitable for large, geographically distributed networks.

d) Access to Diverse Data

By training on data from many different devices, the global model learns to recognize a wider variety of attack patterns than it would from a single, centralized dataset. This makes the final model more robust and generalizable to new, unseen attacks.

e) Reduced Communication Overhead

Instead of transmitting massive raw log files, the devices only send small, compressed model updates. This significantly reduces network bandwidth usage and is ideal for devices with limited or intermittent connectivity.

IV. METHODOLOGY

This section looks at the methodology implemented to undertake this study.

A. Dataset

The Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC) worked together to generate the extensive, thorough, and realistic CSE-CIC-IDS2018 dataset. Setting a standard for assessing intrusion detection systems (IDS) and creating fresh machine learning-based security solutions is its main objective.

Key characteristics

1) Distributed Nature

To replicate the distributed nature of an IoT edge computing environment, the dataset can be readily divided into several sub-datasets from various days and attack kinds. By simulating a local data repository, you can assign a subset of the data to each device in your federated learning model.

2) SSH Brute Force Attacks

By adding labels for certain SSH brute force attacks, we can train the models to identify the precise kind of threat you're looking for.

3) Federated Learning Compatibility

The data is easy to use because it has already been pre-processed into a tabular, feature-rich format. Only the model changes need to be shared, not the data itself, which is entirely consistent with the privacy-preserving tenets of federated learning. The CSE-CIC-IDS2018 dataset has already been utilised in a number of research studies to develop and evaluate federated learning intrusion detection systems.

4) Various Attack Scenarios

Since the dataset includes a particular scenario for brute force attacks, it is ideal for detecting them. There are seven distinct attack categories in it: Denial of Service (DoS), Distributed Denial of Service (DDoS), Brute-force, Heartbleed, Botnet, Web attacks, and network intrusion.

B. Simulation environment

Google Colab
We use Google Colab because it offers a strong, cost-free, and easily available environment for the project's development and simulation stages, Google Colab is extremely pertinent to this investigation. One of its main benefits is free access to powerful hardware, such as GPUs and TPUs, which are necessary for training intricate machine learning models on huge datasets like CSE-CIC-IDS2018. Furthermore, it contains all required libraries pre-installed, enabling rapid prototyping without the trouble of a complicated setup. Above all, by dividing data and implementing code to simulate local client training and asynchronous model aggregation, it is a perfect platform for emulating the distributed nature of a federated learning system in a single environment.

C. Simulation parameters

TABLE I. SIMULATION PARAMETERS

Simulation Parameters	
Simulation Parameter	***Setting***
Network Simulator	Google Colab
Programming language	Python
Operating System	Windows 10 OS
Simulation Area	1000mx1000m
Simulation metrics	Accuracy, Precision, Recall, F1-Score, and False Alarm Rate
Packet Analyser	Wireshark
Mobility Model	Random Waypoint Model (RWM)
Traffic Type	Constant Bit Rate

D. Python libraries used for CH-federated Learning

1) TensorFlow Federated (TFF)

TFF, developed by Google, is specifically designed to simulate FL environments, making it ideal for a Colab setup. Crucially, its low-level API, Federated Core (FC), allows for the definition and customization of the entire communication and aggregation process.

2) TensorFlow/Keras

Its role is to define the actual IDS model that runs locally on each simulated IoT device.

3) Pandas/NumPy

Used for data preprocessing, splitting the original dataset into Non-Independent and Identically Distributed (non-IID) partitions for each "device" (client), and handling model weights/metrics.

4) Scikit-learn

Used for data normalization, feature scaling, and calculating standard performance metrics for the final results.

E. Components to Simulation Logic

The implementation moves from defining the ML model and the dataset to customizing the core FL process, particularly the aggregation.

1) Data Preparation (Cross-Device)

A public IoT-focused network traffic dataset, CSE-CIC-IDS2018 is loaded into the Colab environment.

2) Cross-Device Partitioning

The full dataset is partitioned to simulate the data distribution across many separate IoT devices (clients).

Horizontal FL: All devices have the same feature space (network traffic features) but different data samples.

Non-IID Simulation: The data is usually partitioned to be Non-Independent and Identically Distributed (Non-IID). For example, some simulated clients might have a higher proportion of legitimate traffic, while others have more attack samples, reflecting real-world device heterogeneity. TFF structures this using a `tff.simulation.datasets` object.

3) Model Definition

The model is compiled with an appropriate loss function, Binary Cross-Entropy and an optimizer, SGD.

4) Asynchronous Aggregation Implementation (TFF Customization)

This is the most critical and complex step, moving beyond the standard Synchronous Federated Averaging (FedAvg) built into TFF.

The process is generally implemented as a stateful iterative process using TFF's low-level APIs:

- Iterative Process Definition: Instead of using the default `tff.learning.algorithms.build_weighted_fed_avg`, a custom iterative process is built to manage the server's state, which is crucial for synchronicity.
- Server State: The server state must track not only the Global Model Weights (w_t) but also the time/staleness of the last update received from each client, or a running count of client contributions (N_{total}).
- Client Selection/Timing: The simulation environment must be modified to introduce staggered client completion times. This is typically done by assigning a random, heterogeneous training time to each client's local update.
- Asynchronous Update Logic: When a client submits its update (Δw_i), the server immediately calculates the new global model (w_{t+1}), without waiting for a full cohort. A common asynchronous aggregation formula is used, often involving a staleness-based weighting function (η_i):

$$w_{t+1} = w_t - \eta \sum_{i \in s} \lambda_i \cdot \nabla w_i$$

Where:

η - global learning rate.

S - set of clients whose updates have just arrived.

∇w_i - model update (gradient) from client *i*.

λ_i - dynamic weight that penalizes stale (delayed) updates to mitigate the "old update" problem common in asynchronous FL.

5) Simulation Execution

The custom iterative process is run for a fixed number of communication rounds. Since it's asynchronous, a "round" might be defined as simply the reception and aggregation of one client update, rather than waiting for a full cohort. The metrics are then logged after every aggregation step.

V. RESULTS AND DISCUSSION

This section we look at comparison results. In our results we compare our proposed model with the traditional model called Fail2ban and also the machine learning model called LSTM.

Table II depicts the comparison results for the 3 protocols using 5 different metrics; namely accuracy, precision, recall, f1-score and false alarm rate.

TABLE II. COMPARISON RESULTS

Model	*Accuracy*	*Precision*	*Recall*	*F1 Score*	*False Alarm Rate*
Fail2Ban	0.808602	0.575472	0.809735	0.672784	0.191761
LSTM	0.9	0.929032	0.637168	0.755906	0.015625
CH-Federated Learning	0.916129	0.930233	0.707965	0.80402	0.0170455

A. *Accuracy Results*

Fig.3. illustrates the accuracy comparison results. Fail2Ban's lower performance is due to its rule-based, reactive nature. It works by monitoring log files and banning an IP address only after a predefined number of failed login attempts are reached within a specific time. This simple, static approach is vulnerable to distributed or "slow" brute-force attacks that intentionally stay under the configured threshold. As it lacks the ability to learn from dynamic attack patterns, it can be easily bypassed by sophisticated adversaries who change their IP addresses or spread out their login attempts over a longer period.

The high performance of the proposed model, as well as the LSTM model, is because they are machine learning-based protocols. The LSTM's ability to analyse sequential data allows it to identify subtle, time-based patterns in login attempts, making it more effective than Fail2Ban's static rules. The proposed model further refines this by using asynchronous horizontal federated learning, which enables it to learn collaboratively from a vast, decentralized dataset of IoT devices at the edge without compromising data privacy. This collective intelligence allows the model to become more robust and accurate in detecting complex, real-world attack behaviours that a single model or a rule-based system could never handle.

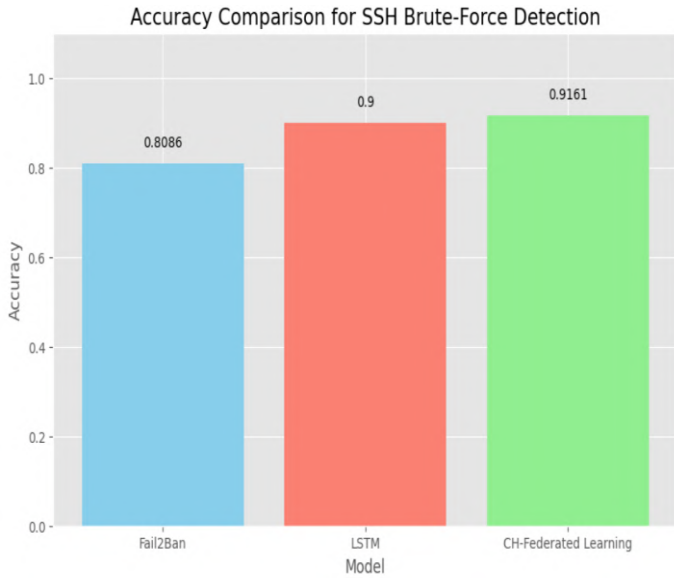


Fig. 3. Accuracy Comparison Results

B. *Precision Results*

Fail2Ban demonstrates a significantly lower precision of approximately 57.55%. This indicates that when Fail2Ban identifies an SSH brute-force attack, there's a relatively high chance that its detection is a false positive – meaning it incorrectly flagged legitimate login attempts as malicious. This low precision can lead to legitimate users being temporarily banned, causing inconvenience and potential service disruptions due to its static, rule-based nature.

The proposed CH-Federated Learning model achieves the highest precision. In a distributed edge computing environment, it allows each IoT device to train its own local model using its private data. The central server then aggregates these local model updates asynchronously to create a more robust global model. This approach effectively pools the collective intelligence of all devices, enabling the model to learn from a much wider range of attack behaviours without ever accessing raw data, thereby ensuring both high accuracy and data privacy, and resulting in superior performance.

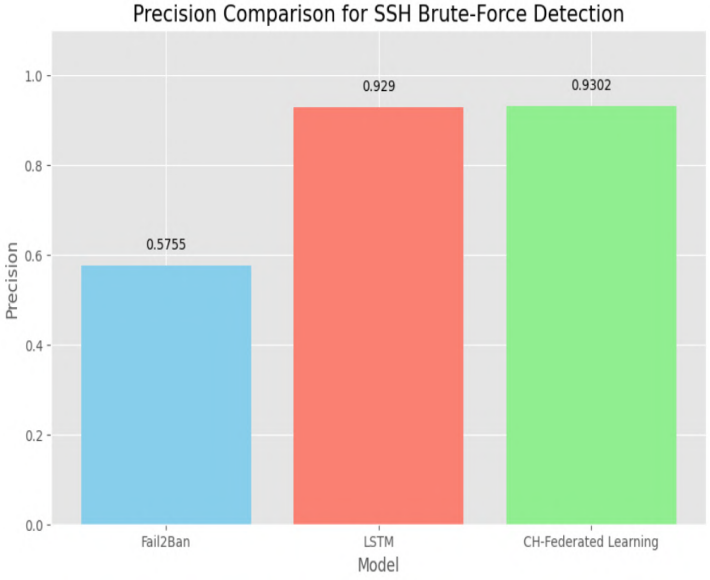


Fig. 4. Precision Comparison Results

C. *Recall Results*

Fail2Ban's relatively high recall is a byproduct of its broad, rule-based detection mechanism. By setting simple thresholds, such as banning an IP after a few failed login attempts, it effectively casts a wide net. This approach tends to catch a larger percentage of actual brute-force attempts, as it doesn't require complex patterns to be met. However, this high recall often comes at the cost of precision, meaning it is more prone to mistakenly flagging legitimate users (as seen in the high false alarm rate). It prioritizes identifying any suspicious activity, ensuring fewer actual attacks are missed, even if some of its alerts are false.

In contrast, the LSTM model shows a lower recall of approximately 63.72%. This suggests that while LSTM has high precision (as seen previously), it might miss a noticeable percentage of actual brute-force attacks. The proposed CH-Federated Learning model achieves a recall of 70.8%, which is better than the standalone LSTM but still lower than

Fail2Ban. This implies that while the machine learning models excel at minimizing false positives (high precision), their nuanced detection mechanisms, especially when balancing against precision, might not capture every single attack instance as broadly as Fail2Ban's more general rules. The trade-off between precision and recall is evident here, where the machine learning models prioritize the accuracy of positive detections, potentially at the cost of missing a few more actual attacks compared to a system that casts a wider net

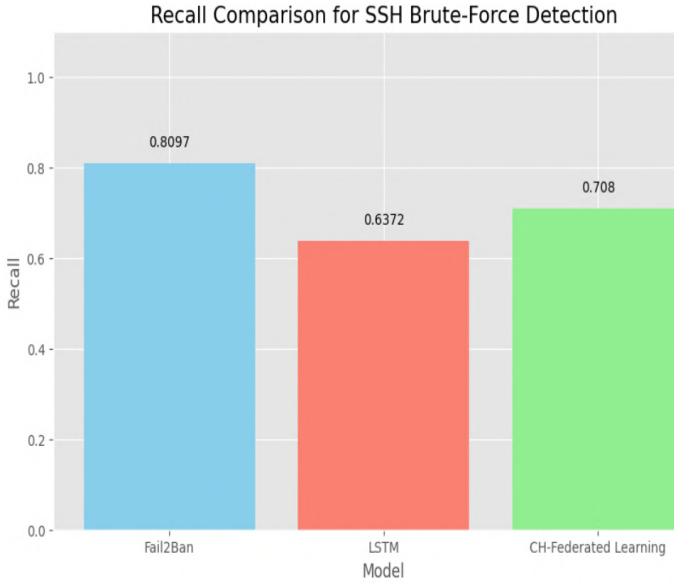


Fig. 5. Recall Comparison Results

D. *F1 Score Results*

The F1 score provides a balanced measure of a model's performance, considering both precision and recall. Fail2Ban's F1 score of 0.6728 is relatively low because its decent recall is heavily counteracted by its very low precision. The simplicity of its rule-based detection means it often misidentifies legitimate actions as threats, leading to a high number of false positives that significantly drag down its overall balanced performance, despite its ability to broadly detect suspicious activities.

The proposed CH-Federated Learning model achieves the highest F1 score of 0.804 by optimally balancing both precision and recall, demonstrating its superior overall effectiveness. It combines the advanced pattern recognition of a machine learning model, similar to LSTM, with the benefits of distributed learning across multiple IoT devices. This approach allows it to leverage a wider and more diverse dataset of attack behaviours without centralizing raw data, leading to a more robust and generalized model that enhances both its ability to accurately detect attacks (high precision) and its capacity to identify a broad range of them (improved recall).

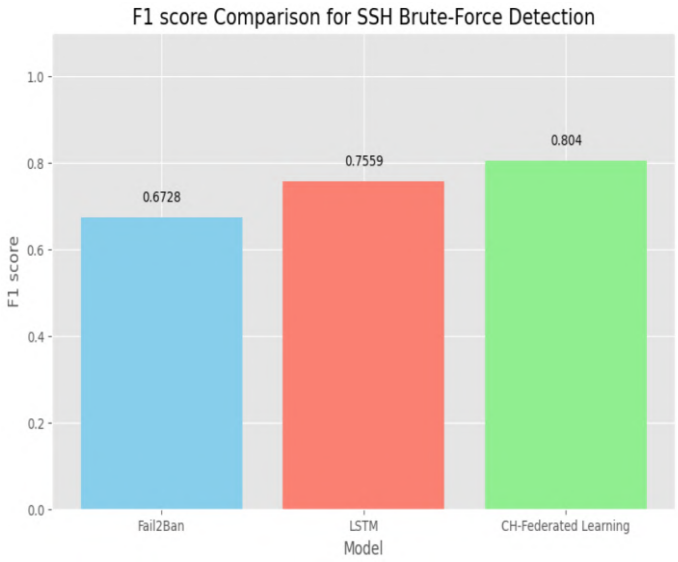


Fig. 6. F1 Score Comparison Results

E. *False alarm rate Results*

Fail2Ban exhibits a significantly high false alarm rate. Fail2Ban's high false alarm rate is a direct consequence of its static, rule-based approach. It operates on a simple principle: if a certain number of failed login attempts occur within a set timeframe from a single IP address, that IP is blocked. This rigid rule-set cannot distinguish between a malicious actor and a legitimate user who simply mistyped their password multiple times. As a result, it often bans legitimate users, leading to a high rate of false positives.

In stark contrast, both the LSTM and the proposed CH-Federated Learning models demonstrate remarkably low false alarm rates. The LSTM and proposed CH-Federated Learning models use machine learning to analyse complex patterns. Instead of relying on a simple count, they learn to differentiate between normal and malicious behaviour by considering a wide range of factors, such as the timing between login attempts, the usernames used, and other sequential data. This nuanced understanding of behavioural patterns allows them to accurately classify an event as a real attack, even when a user might have multiple failed attempts. The high precision and low false alarm rate of these models demonstrate their ability to avoid the pitfalls of a rigid, rule-based system, making them far more reliable for real-world application.

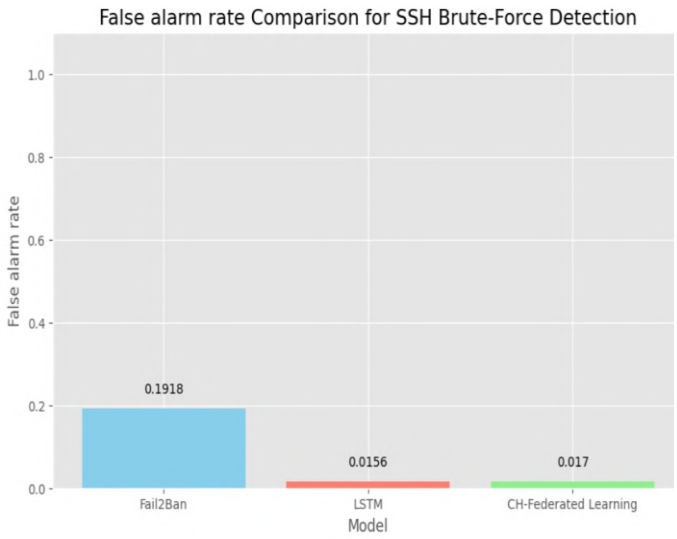


Fig. 7. False alarm rate Comparison Results

VI. Conclusion

In conclusion, this research successfully demonstrated the efficacy of a CH-Federated Learning framework for detecting Secure Shell (SSH) Brute Force attacks within the unique challenges of an Internet of Things (IoT) edge computing environment. By leveraging an asynchronous-based aggregation mechanism, our approach addressed critical issues of privacy, scalability, and communication efficiency inherent in distributed networks. The experimental results, validated using the comprehensive dataset, confirmed that the proposed framework achieves high detection results while allowing IoT devices to train models collaboratively without exposing sensitive local data. This work provides a significant contribution to the field of IoT security by offering a robust and practical solution that can be deployed across a wide range of devices, paving the way for more resilient and secure edge ecosystems. Future work should explore the integration of a trust mechanism to handle potential malicious clients and investigate the framework's performance on other types of network attacks.

References

[1] T Alam, "Computing and its role in the Information Technology," *IAIC Trnsaction on Sustainable Digital Innovation (ITSDI)*, vol. 1, no. 2, April 2020.

[2] A S Almogren, "Intrusion detection in Edge-of-Things computing," *Journal of Parallel and Distributed Computing*, vol. 137, pp. 259-265, 2020.

[3] Wei Yu et al., "A Survey on the Edge Computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900-6919, 2017.

[4] M M Raikar and S M Meena, "SSH brute force attack mitigation in Internet of Things (IoT) network : An edge device security measure," *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*, pp. 72-77, 2021.

[5] D Chen et al., "S2M: A lightweight acoustic fingerprints-based wireless device authentication protocol," *IEEE Internet Things Journal*, vol. 4, no. 1, pp. 88-100, 2016.

[6] D Chen et al., "Physical layer based message authentication with secure channel codes," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 1079-1093, 2018.

[7] H Xue, D Chen, N Zhang, H N Dai, and K Yu, "Integration of blockchain and edge computing in internet of things: A survey," *Future Generation Computer Systems*, vol. 144, pp. 307-326, July 2023.

[8] I Sittón-Candanedo, "Edge computing: A review of application scenarios," *International Symposium on Distributed Computing and Artificial Intelligence*, pp. 197-200, 2019.

[9] N Tiwari and N Hubballi, "Secure Socket Shell Bruteforce Attack Detection With Petri Net Modeling," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 697-710, 2023.

[10] A Z Agghey, L J Mwinuka, S M Pandhare, M A Dida, and J D Ndibwile, "Detection of Username Enumeration Attack on SSH Protocol: Machine Learning Approach," *Symmetry*, vol. 13, 2021.

[11] R George and E Zeratsion, "Detection of SSH Brute-Force Attacks Using Machine Learning," University West, Department of Engineering Science, Trollhattan, SWEDEN, 2025.

[12] A Tely, A Aryanti, and S Soim, "Sharing SSH Threat Intelligence across Multiple Servers using WebSocket and Fail2Ban," *Information Technology Engineering Journals*, vol. 10, no. 2, pp. 221-229, 2025.

[13] G S Lab. (2021) Fail2exploit: A security audit of fail2ban. [Online]. https://securitylab.github.com/research/Fail2exploit/, 2021

[14] KeyCDN. (2021, July) Fail2ban – the python based and popular ips framework. [Online]. https://www.keycdn.com/support/fail2ban, 2021

[15] S K Wanjau, G M Wambugu, and G N Kamau, "SSH Brute Force Attack Detection Model based on Deep Learning," *International Journal of Computer Applications Technology and Research*, vol. 10, pp. 42-50, 2021.

[16] M D Hossain, H Ochiai, F Doudou, and Y Kadobayashi, "SSH and FTP brute-force attacks detection in Computer Networks: LSTM and Machine Learning Approaches," *2020 5th International Conference on Computer and Communication Systems (ICCCS)*, 2020.

[17] A Hamza, H Khalid, T N Usmani, M Z Hussain, and M Z Hasan, "SSH ATTACKS DETECTION USING MACHINE LEARNING: COMPARATIVE ANALYSIS OF DIFFERENT ML MODELs," *Spectrum of Engineering Sciences*, vol. 3, no. 8, pp. 903-915, 2025.

[18] M M Najafabadi, T M Khoshgoftaar, C Kemp, N Seliya, and R Zuech, "Machine learning for detecting brute force attacks at the network level," *2014 IEEE International Conference on Bioinformatics and Bioengineering*, 2014.

[19] D M Atallah, M Badawy, A EI-Sayed, and M Ghoneim, "“Predicting kidney transplantation outcome based on hybrid feature selection and KNN classifier," *Multimedia Tools and Applications*, vol. 78, no. 14, pp. 20383–20407, 2019.

Network Operations and Management; Network Standards and Regulatory

A Comparison of IEEE 1900.5.2 Spectrum Consumption Models with Conventional Methods for Coexistence Analysis

1st Schuyler Scholtz, 2nd Melvin Ferreira, 3rd Herman Blackie
School of Electrical, Electronic and Computer Engineering
North-West University
Potchefstroom, South Africa
31696309@mynwu.ac.za[1], melvin.ferreira@nwu.ac.za[2], herman.blackie@nwu.ac.za[3]

***Abstract*—Conventional spectrum coexistence methods often impose over-conservative protection criteria for incumbent services, resulting in inefficient use of the limited radio frequency (RF) spectrum. IEEE 1900.5.2 Spectrum Consumption Models (SCMs) facilitate Dynamic Spectrum Management (DSM) to optimise spectrum utilisation with greater flexibility. This study compares IEEE 1900.5.2 SCMs with a conventional coexistence method from the ITU-R S.2199 report to model opportunities for spectrum sharing. We detail both coexistence strategies and apply them to a scenario where a Broadband Wireless Access (BWA) base station seeks to share the C-band spectrum with a Fixed-Satellite Service (FSS) earth station. The IEEE 1900.5.2 SCMs outperform the conventional approach in co-channel situations but initially offers fewer sharing opportunities in the first and second adjacent channels. Additionally, IEEE 1900.5.2 SCMs can assist in identifying more spectrum sharing opportunities within the elevation angles commonly used by FSS earth stations in South Africa.**

***Index Terms*—Broadband Wireless Access (BWA), Coexistence, Dynamic Spectrum Management (DSM), Fixed-Satellite Service (FSS), IEEE 1900.5.2, Interference analysis, Spectrum Consumption Model (SCM), Spectrum compatibility, Spectrum sharing.**

I. Introduction

The radio frequency (RF) spectrum is composed of a finite range of frequencies that require coordinated management for global radio communication connectivity. Conventional spectrum management methods often use a link budget equation to determine if the interfering power from a transmitter is greater than the interference threshold set by a victim receiver. Various discrete values are added and subtracted to ultimately determine if an inequality is met and coexistence is possible. The main drawback of this static approach involves simplified assumptions that do not account for network mobility and schedule variability across target services. Additionally, the limited consideration of receiver characteristics, such as a fixed interference threshold that does not account for receiver filtering, may lead to misestimating interference impacts. Conventional coexistence methods typically set overly conservative protection standards for Primary Users (PUs), enforcing strict minimum separation distances that Secondary Users (SUs) must maintain.

With the accelerating growth of wireless services and devices, conventional methods fall short, necessitating a transition to Dynamic Spectrum Access (DSA) to improve utilisation of the limited RF spectrum [1]. Dynamic Spectrum Management (DSM) is one DSA approach [1], enabling SUs to access spectrum allocated to PUs when they are dormant. Additionally, spectrum sharing is permitted if SUs ensure interference does not disrupt the PUs' operations [2]. Current DSA-based systems, such as the Citizens Broadband Radio Service (CBRS) and Television White Space (TVWS), rely on a geo-location database with precomputed values to determine coexistence among services.

Spectrum Consumption Models (SCMs) were introduced in [3] as a new approach to facilitate DSM. The concept allows stakeholders to exchange only a minimal amount of information that is required to determine coexistence in shared spectrum, whilst remaining independent of each other's detailed operations. An SCM describes spectrum usage in time, frequency, and spatial dimensions.

SCMs evaluate the compatibility between transmitters and receivers of different services by specifying energy radiation and permissible interference levels to prevent harmful interference. SCMs are self-contained and do not require an external database to function, as they facilitate interoperability among complex systems through a practical data exchange specification between neighbouring services. A computationally efficient compatibility evaluation determines whether services can coexist, allowing fast decision outputs for granting or denying spectrum resources to SUs. The IEEE Dynamic Spectrum Access Networks Standards Committee (DySPAN-SC) 1900.5 Working Group has formalised the process for creating, exchanging, and evaluating SCMs to

This work is based on the research supported in part by the National Research Foundation (NRF) of South Africa (Ref. Number FTPM250224299748). Any opinion, finding, conclusion, or recommendation expressed in this material is that of the author(s) alone, and the NRF accepts no liability whatsoever in this regard.

resolve spectrum compatibility between homogeneous and heterogeneous services [4]. The broad adoption of SCMs aims to address numerous shortcomings present in conventional coexistence methods, ultimately leading to more efficient spectrum use.

The C-band (3400 - 4200 MHz) offers a reliable solution for balancing network speed and coverage in spectrum applications. The band is primarily occupied by Fixed-Satellite Services (FSSs) operating as PUs. However, given the desirable characteristics of this band, many regions are starting to co-locate Broadband Wireless Access (BWA) services as SUs alongside the FSSs. This causes harmful interference to FSS earth stations, necessitating spectrum coexistence methods to reduce the risk of PUs' service disruption [5].

The norm for determining compatibility between FSS earth stations and BWA base stations includes applying a conventional spectrum coexistence method. However, this approach introduces missed opportunities in spectrum usage due to the drawbacks previously mentioned.

To our knowledge, there are no related works that compare IEEE 1900.5.2 SCMs with conventional spectrum coexistence methods. In this paper, we address this gap by demonstrating the effectiveness of IEEE 1900.5.2 SCMs compared to a widely adopted conventional coexistence method in modelling spectrum coexistence opportunities. We employ a subset of the ITU-R S.2199 report [5] as a comparative case study, due to its significance in evaluating coexistence between an FSS earth station and a BWA base station functioning in the C-band.

The relative results from [5] are replicated by first utilising the parameters specified in the report with input variation, and subsequently using IEEE 1900.5.2 SCMs. Ultimately, the study aims to demonstrate the usefulness of applying SCMs as opposed to the conventional method used in ITU-R S.2199 [5].

The rest of the paper is organised as follows: In Section II, we discuss related work. Sections III and IV discuss the conventional coexistence and IEEE 1900.5.2 SCMs, respectively. Section V explains the method and the evaluated scenario. The results are discussed in Section VI, and Section VII concludes the paper.

II. Related Work

A. Spectrum Consumption Models

The construction of an open source SCM application, Spectrum Consumption Model Build and Analysis Tool (SCMBAT), was presented and evaluated in [6]. The purpose of SCMBAT is to encourage the adoption of SCMs, contributing to the advancement of spectrum sharing.

In [7], a wireless experimentation framework is developed on the National Science Foundation (NSF) PAWR COSMOS testbed. The framework was designed to investigate and demonstrate automated DSA coordination via SCMs. An experiment involving three homogeneous networks, each entering the wireless communication environment sequentially, was performed. By exchanging SCM messages, the newly joined network could autonomously identify and adjust to frequencies that would not cause harmful interference based on the transmission characteristics of the existing peer networks.

Following from [7], a novel SCM-based DSA algorithm, was developed in [8]. The algorithm seeks to resolve spectrum conflicts by minimising interference with existing devices and systems, achieved through the dynamic adjustment of RF operational parameters for a transmitter-receiver pair. Two methods for using the algorithm were also introduced: sequential and distributed. The sequential method considers total interference from all transmitters at each receiver, while the distributed method evaluates interference peer-to-peer. The distributed DSA method showed an increase in the speed of spectrum deconfliction by up to seven times compared to the sequential method in large-scale scenarios.

B. Fixed-Satellite Service

The work in [9] aimed at evaluating the impact 5G cellular systems have on FSS earth stations in both co-channel and adjacent channel emissions. Results indicated that co-channel interference is significantly greater than interference from adjacent channel emissions when assessing individual base stations. Aligning the FSS earth station antenna was crucial to achieving varied separation distances between a base station and earth station, with misaligned antennas needing less distance to avoid harmful interference. To promote coexistence, it is suggested to turn off base stations with high interference. Alternatively, base stations can be grouped by similar interference levels, and each group's transmission power can be reduced.

A stochastic geometry-based analytical framework for assessing spectrum coexistence between terrestrial and satellite services was proposed in [10], addressing both co-channel and adjacent channel interference. The framework derives limits on location-based base station density, maximum transmission power, and Spectrum Emission Mask (SEM) requirements for terrestrial transmitters to ensure no harmful interference towards an FSS service.

Despite suggestions made from the related works within this section to facilitate spectrum coexistence between terrestrial and satellite networks, none of them utilised SCMs. Prioritising an effective comparison between SCMs and conventional spectrum coexistence techniques, it is essential to use an established coexistence analysis as a reference. For this reason, we have selected the ITU-R S.2199 report [5] as the basis for our case study.

III. Conventional Coexistence Analysis

The conventional coexistence method employed by the ITU-R S.2199 report [5] is detailed in this section. The method involves a conditional inequality that should be met to indicate service compatibility between the interfering transmitter and the victim receiver. Refer to (1), which is derived from the FSS earth station receiver interference tolerance.

$$\begin{aligned} PL(d) - AP_{\mathrm{TX}}(d) - AP_{\mathrm{RX}}(d) \geq P_{\mathrm{TX}} + G_{\mathrm{TX,Max}} \\ - L_{\mathrm{TX,FL}} + G_{\mathrm{RX,Max}} - ACLR - L_{\mathrm{p}} - I_{\mathrm{Tolerance}} \end{aligned} \quad (1)$$

TABLE I
ITU-R S.2199 EQUATION PARAMETERS FOR (1), FROM [5]

Parameter	Description	Unit
d	Distance between transmitter and receiver	m
$PL(d)$	Path loss from transmitter to receiver a distance, d	dB
$AP_{\mathrm{TX}}(d)$	Transmitter's normalised antenna pattern	dB
$AP_{\mathrm{RX}}(d)$	Receiver's normalised antenna pattern	dB
P_{TX}	Maximum transmit power	dBW
$G_{\mathrm{TX,Max}}$	Maximum antenna gain of transmitter	dBi
$L_{\mathrm{TX,FL}}$	Feeder/Insertion loss of transmitter	dB
$G_{\mathrm{RX,Max}}$	Maximum antenna gain of receiver	dBi
$ACLR$	Adjacent channel leakage ratio of transmitter	dB
L_{p}	Penetration loss	dB
$I_{\mathrm{Tolerance}}$	Maximum allowable interference of receiver	dBW

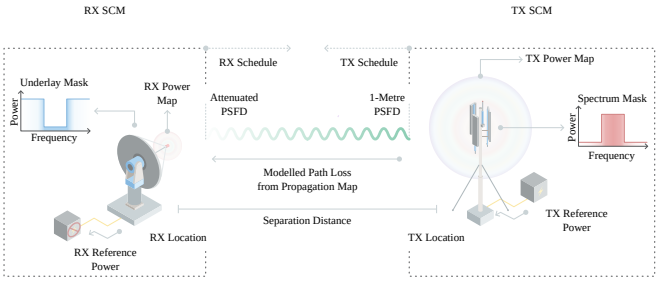


Fig. 1. IEEE 1900.5.2 Process to build a complete Spectrum Consumption Model. TX is the interfering transmitter, a BWA base station, and RX is the victim receiver, an FSS earth station, respectively.

Each parameter is defined in Table I. The ratio of power leaking into adjacent channels compared to the power present within the intended channel is given by the Adjacent Channel Leakage Ratio (ACLR). Any building or structural penetration loss is included within L_{p}.

In this instance, the BWA base station is the interfering transmitter, and the FSS earth station is the victim receiver.

Rearranging (1) for $PL(d)$ yields the inequality in (2). $PL(d)$ captures the minimum path loss required between the BWA base station transmitter and FSS earth station receiver to ensure service compatability and coexistence.

$$\begin{aligned} PL(d) \geq\ & P_{\mathrm{TX}} + G_{\mathrm{TX,Max}} + AP_{\mathrm{TX}}(d) - L_{\mathrm{TX,FL}} \\ & + G_{\mathrm{RX,Max}} + AP_{\mathrm{RX}}(d) - ACLR - L_{\mathrm{p}} - I_{\mathrm{Tolerance}} \end{aligned} \tag{2}$$

Finally, the interference tolerance, $I_{\mathrm{Tolerance}}$, pertains to the long-term maximum permissible interference the victim receiver can tolerate. In this instance, the FSS earth station can tolerate this interferer 20% of the time [5]. The interference tolerance is also referred to as the interference temperature model [11], and is mathematically defined in (3) as used in [5], based on [12].

$$P_r(20\%) = 10 \cdot \log(k \cdot T_r \cdot B) + J - W \tag{3}$$

where the power level, $P_r(20\%)$, is measured in dBW and $I_{\mathrm{Tolerance}} = P_r(20\%)$. Here, k is Boltzmann's constant, T_r signifies the noise temperature of the receiving system in K, and B is the reference bandwidth in Hz. Moreover, J represents the ratio between acceptable long-term interference from an individual source and the thermal noise power in the receiving system in dB. W is the measure of equivalent thermal noise for interfering emissions in the reference bandwidth, also in dB. A positive W value suggests that interference noise has a greater adverse effect compared to thermal noise. The values of both J and W are available in [12].

IV. IEEE 1900.5.2

Evaluating spectrum compatibility with IEEE 1900.5.2 [4] starts by defining a set of constructs for the interfering transmitter (BWA base station), and the victim receiver (FSS earth station). These constructs collectively characterise the spectrum consumption of the respective service in the form of an SCM. The seven constructs used in this study are described below, with Fig. 1 providing a high-level view of the entire process to build a complete SCM with IEEE 1900.5.2.

1) Reference Power: Baseline power level in dBW. For a transmitter, it is the power that drives the antenna. In receivers, it is the total interference tolerance after frequency- and time-dependent rejection.

2) Spectrum Mask: Also known as the SEM. Defines RF emission bounds for a transmitter in $\mathrm{dB/Hz}$ when unscaled by any other constructs. The resolution bandwidth (RBW) refers to the width of the frequency band over which the analyser measures and integrates power.

3) Underlay Mask: Specified for the receiver and indicates the permissible Power Spectral Density (PSD) in $\mathrm{dB/Hz}$ from distant interferers when unscaled by any other constructs. A RBW is once again included as in SEMs.

4) Power Map: Models the distribution (transmit) and accumulation (receive) of electromagnetic energy as directional Power Flux Density (PFD) in $\mathrm{dB/m^2}$, relative to the reference power at one metre.

5) Propagation Map: Attenuation rate of a transmitted signal in dB by direction. The standard includes two variants of the log-distance pathloss model: a linear variant with one exponent and a piecewise linear variant with two exponents and a breakpoint specified in metres. The latter accounts for abrupt changes in path loss using a different exponent after the specified breakpoint. Higher exponents indicate more signal attenuation due to increased destructive interference from the environment.

6) Location: Coordinates of transmitters and receivers in latitude, longitude, and altitude above ground level (AGL). These determine the Angle of Departure (AoD) from the transmitter or Angle of Arrival (AoA) at the receiver, both in degrees, and separation distance in metres.

7) Schedule: Defines the on and off times for spectrum use in Coordinated Universal Time (UTC).

A. Compatability assessment

A series of computational steps is done to assess the compatibility between an interfering transmitter and a victim receiver based on the defined constructs [4]. The initial

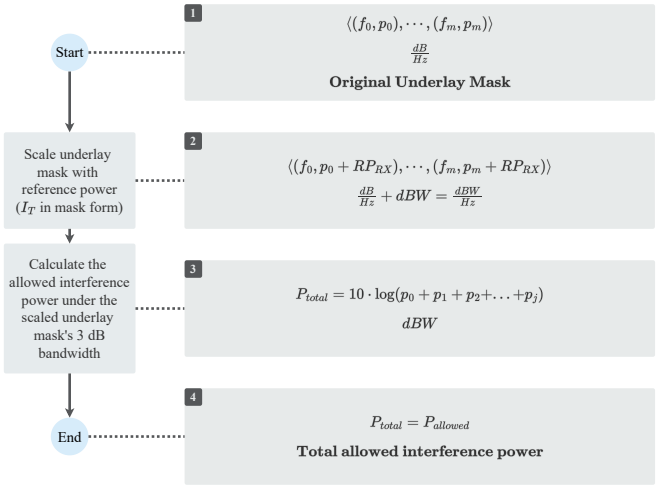


Fig. 2. Receiver-Side Power Margin Calculations

step is to verify whether there is a frequency or schedule overlap between the two services' SCMs. In the absence of an overlap, the models are compatible. However, if an overlap is present, additional computations are required. The compatibility between the two services is quantified by a power margin (PM). The PM is defined as the difference between the *total interference power* as perceived at the victim receiver, and the receiver's *allowed interference power* [4].

IEEE 1900.5.2 outlines two methods for determining the PM: the total power method and the maximum power density method. The total power method is used in this paper.

B. Allowable Interference Power

The maximum allowable interference power from a single interfering transmitter at the victim receiver corresponds to the steps in Fig. 2.

In step 1, the original underlay mask is presented in the form $\langle (f_0, p_0), \cdots, (f_m, p_m) \rangle$, where $\langle f_0, \cdots, f_m \rangle$ are the frequencies in Hz and $\langle p_0, \cdots, p_m \rangle$ the power terms in $\mathrm{dB/Hz}$, since a RBW is included.

In step 2, the original underlay mask's power values, $\langle p_0, \cdots, p_m \rangle$, are scaled according to the specified reference power, RP_{RX} dBW, by adjusting each power value producing a newly scaled underlay mask, denoted as the interference threshold, I_T in mask form, with the power units $\mathrm{dBW/Hz}$ upon adjusting to a common RBW.

The total power enclosed within the scaled underlay mask's 3 dB bandwidth is calculated in step 3 using integration. Consider the adjacent inflexion point pairs of frequency and power given by (f_1, p_1) and (f_2, p_2). Taking into account $(f_1 < f_2)$, the line equation is given by $(p = b_0 + b_1 f)$, where $(b_1 = \frac{p_2 - p_1}{f_2 - f_1})$ and $(b_0 = p_1 - b_1 f_1)$. The total power within each segment between f_a and f_b is given by (4).

$$p = \int_{f_a}^{f_b} \frac{10^{\frac{b_0 + b_1 f}{10}}}{RBW} df \text{ for } f_1 \leq f_a < f_b \leq f_2 \tag{4}$$

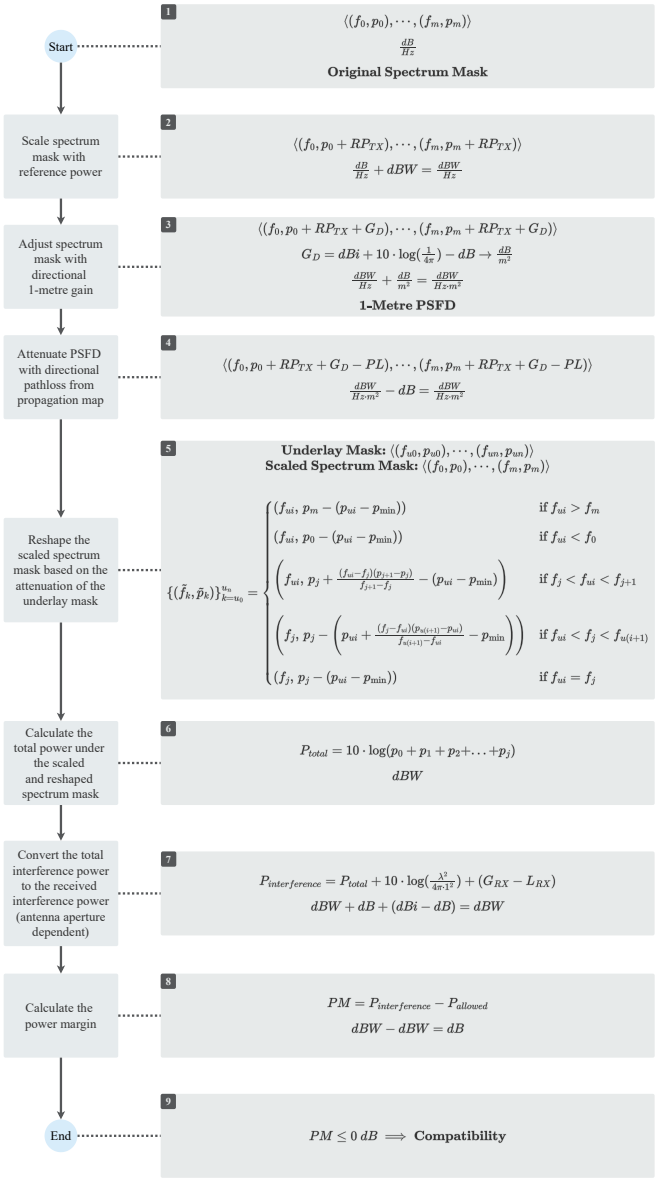


Fig. 3. Transmitter-Side Power Margin Calculations

where RBW is the resolution bandwidth in Hz. The three segments in which computations can take place for $f_1 \leq f_a < f_b \leq f_2$ are defined in IEEE 1900.5.2. each with their respective closed-form equations to compute the power, $\{p_0, p_1, p_2, ..., p_j\}$ in each segment of the mask.

In step 3, the total amount of interference power the receiver can tolerate is calculated (5).

$$P_{total} = 10 \cdot \log\left(p_0 + p_1 + p_2 + ... + p_j\right) \tag{5}$$

where $P_{allowed} = P_{total}$ in dBW is the *allowed interference power* as denoted in step 4.

C. Total Interference Power

Next, a link budget equation is used to determine the *total interference power* from the interfering transmitter, as shown in Fig. 3.

Both location constructs for the BWA base station and FSS earth station are first used to determine the AoD, AoA, and separation distance between the transmitting and receiving antennas.

The original spectrum mask's (SEM) power values, $\langle p_0, \cdots, p_m \rangle$, from step 1 in Fig. 3, is given in $\mathrm{dB/Hz}$, accounting for a RBW. In step 2, the SEM is adjusted by increasing the power values by the specified reference power, RP_{TX} dBW, resulting in a new scaled spectrum mask with the power units $\mathrm{dBW/Hz}$ after adjusting to a common RBW.

In step 3, the AoD is used to determine the directional gain, G_D $\mathrm{dB/m^2}$, from the power map that should be used towards the FSS earth station's location. The scaled spectrum mask's power terms from step 2 are adjusted by the directional antenna gain, a conversion factor, and any insertion losses to represent the Power Spectral Flux Density (PSFD) on a 1-metre isotropic sphere in $\mathrm{dBW/(Hz \cdot m^2)}$. The result is the far-field propagation result, and is known as the maximum field strength of the transmitted signal.

In step 4, the AoD is used to identify the suitable pathloss equation in the propagation map. The separation distance is used in the chosen pathloss equation to determine the signal attenuation, PL dB. This calculated attenuation is then used to adjust the spectrum mask from step 3. The result is the diminished PSFD in $\mathrm{dBW/(Hz \cdot m^2)}$, witnessed at the receiver's side. The diminished PSFD is also referred to as the *transmitted field strength*.

In step 5, the underlay mask serves as a filter that alters the transmitted field strength. Consider the underlay mask, defined as $\langle (f_{u0}, p_{u0}), \cdots, (f_{un}, p_{un}) \rangle$, which may include any arbitrary control point (f_{ui}, p_{ui}), along with the mask's minimum power term, p_{min}. Additionally, the scaled spectrum mask from step 4 is represented as $\langle (f_0, p_0), \cdots, (f_m, p_m) \rangle$, with any arbitrary control point (f_j, p_j) included. The arbitrary control points are utilised to establish the position of a specified point on the spectrum mask in relation to any point on the underlay mask.

The spectrum mask is reshaped based on a set of conditions related to an indexed control point's frequency value, f_j. This yields a new mask, the *reshaped spectrum mask*, spanning the total bandwidth of the underlay mask with the frequencies $\tilde{f}_k$ and adjusted power terms $\tilde{p}_k$ based on the attenuation of the underlay mask. The power terms still hold the unit $\mathrm{dBW/(Hz \cdot m^2)}$.

In step 6, the total amount of interference power, P_{total}, beneath the reshaped spectrum mask is computed. The process and equations used are the same as step 3 in Section IV-B.

In Step 7, a conversion factor is added to the total interference power under the reshaped spectrum mask to represent the electromagnetic energy accumulated by the receiver's antenna aperture. Additionally, the directional antenna gain, G_{RX} of the receiving antenna, determined by the AoA from the transmitter to the receiver, contributes to the total interference power. Any insertion loss, L_{RX} at the receiver, is also accounted for. The result is the *total interference power*, $P_{interference}$.

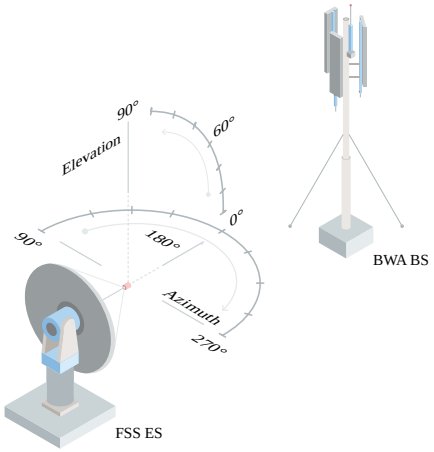


Fig. 4. FSS earth station (ES) orientation relative to interfering BWA base station (BS)

The PM, as defined in Section IV-A, is computed in step 8.

In step 9, the compatibility assessment is performed. A PM value less than or equal to 0 dB indicates compatibility between the two SCM models. Conversely, a power margin greater than 0 dB indicates an incompatible pair, and the services can not coexist.

V. Methodology

A. Scenario

Consider Fig. 4. A cellular urban BWA base station deployment is located some distance away from an FSS earth station. The BWA base station and FSS earth station are directed at each other at a fixed azimuth, since the 3 dB beamwidth of the transmitter is omnidirectional.

The simulation initially positions the BWA base station and FSS earth station 100 m apart, with an initial earth station antenna elevation pointing angle of 5°. If incompatibility is detected, the separation distance is incrementally increased while maintaining the elevation pointing angle constant. The specific elevation angle and the shortest distance indicating compatibility are documented. Both the conventional coexistence and IEEE 1900.5.2 methods are used to assess compatibility.

Following the determination of compatibility, the distance is reset to 100 m, and the elevation pointing angle is incremented by one degree. This procedure is reiterated to find the minimum compatible distance for each earth station antenna elevation pointing angle from 5° to 85°. This procedure is repeated for the co-channel, the first- and second-adjacent channels in the frequency domain.

TABLE II
BWA BASE STATION OPERATING PARAMETERS [5]

Parameter		Value	Unit
Centre frequency		3600	MHz
Channel bandwidth		7	MHz
Peak transmit power		2	dBW
Feeder/Insertion loss		3	dB
Peak antenna gain		9	dBi
Antenna 3 dB beamwidth		0-360	degrees
Antenna downtilt		4	degrees
Antenna centre height AGL		15	m
Unwanted Emissions	ACLR 1	51	dB
	ACLR 2	87	dB

TABLE III
FSS EARTH STATION OPERATING PARAMETERS [5], [15]

Parameter		Value	Unit
Centre frequency		3600	MHz
Channel bandwidth		7	MHz
Antenna diameter		8	m
Feeder/Insertion loss		0	dB
Peak antenna gain		47.7	dBi
Antenna centre height AGL		5	m
Total Noise temperature		70	K
Antenna elevation range		5-85	degrees
Long-term maximum permissible interference (20% of time)		151.7	dBW
Selectivity	ACS 1	45	dB
	ACS 2	50	dB

B. Parameters

Refer to Table II and Table III for a summary of the respective operating parameters as used in this paper. The parameters are from the ITU-R S.2199 report [5].

All variables defined in (1) are first determined to evaluate compatibility accordingly. The ITU-R P.452-13 [13] propagation model is used to determine path loss in the ITU-R S.2199 report [5]. In this work, we used the ITU-R P.452-18 implementation [14]. Version 18 yielded different path loss curves compared to version 13 for similar input parameters. We suspect these differences are due to changes in how clutter loss is handled between the versions. The method of diffraction loss modelling also changed from a Deygout model in version 13 to a delta-Bullington model in version 18 of the the ITU-R P.452 model. To compare our new results with the orginal results in ITU-R S.2199, denoted by *ITU S.2199 Copy*, an extra 7 dB attenuation was added to the version 18 path loss curves.

C. Conventional Coexistence

The scenario and parameters provided above are sufficient to perform a conventional coexistence analysis as described in Section III. Input variation implies that some of the input parameters in (1) will be varied and the resulting output observed to determine the minimum separation distances for compatibility [16].

In this scenario, separation distance, frequency separation and antenna elevation pointing angles will be varied. The separation distance will be increased per FSS earth station elevation pointing angle until the inequality in (1) is met.

D. Coexistence with IEEE 1900.5.2

The IEEE 1900.5.2 SCM method was detailed in Section IV. The original spectrum mask is normalised to a common RBW. Any RBW may be used for the masks, as long as they are normalised to the same RBW. For this paper, the common RBW follows from recommendation ITU-R SM.443-4 [18] and is shown in (6).

$$RBW_{common} = (1.5 \cdot BW_{emission}) \cdot 1\% \tag{6}$$

where $BW_{emission}$ is the estimated bandwidth of the transmitted signal. Once RBW_{common} is computed, the spectrum mask's PSD values are individually normalised to this RBW value via (7).

$$PSD_{new} = PSD_{old} + 10 \cdot \log\left(\frac{RBW_{common}}{RBW_{old}}\right) \tag{7}$$

where PSD_{new} is the new PSD value for a given frequency, PSD_{old} the old PSD value for the same frequency, and RBW_{old} the original RBW value specified for the given frequency, dependent on the offset from the centre frequency. This operation collectively yields an equivalent spectrum mask with a common RBW and continuous PSD values, as is shown in Fig. 5a.

The antenna radiation pattern, along with any insertion and spatial spreading losses, defines the BWA base station power map. The antenna radiation pattern is provided in ITU-R F.1336-2 [19] and the insertion loss is provided in Table II. The radiation pattern is originally computed as antenna gain values in dBi per elevation angle ranging from -90 to 90 degrees. Each antenna gain value is spread across the full 0 to 360-degree azimuth range since the antenna is a monopole and thus omnidirectional.

Given that elevations span from -90 to 90 degrees and azimuths extend from 0 to 360 degrees, these angle conventions conform to the IEEE 1900.5.2 convention, thus negating the need for any angle conversion. However, to adhere to the format of SCM maps as given in [4], each gain value should be bounded by two elevations across a full 0 to 360-degree azimuth range. Bounding by elevation involves iterating over the specified elevations, designating the starting elevation from a particular index and the ending elevation as the subsequent one. The sectoral gain is identified as the maximum gain between these consecutive elevations. Each sector thus has the format $(\phi_i, 0°, G_{dBi}, 360°, \phi_{i+1})$ and includes elevation pairs ranging from -90 to 90 degrees. Fig. 5c shows the BWA base station's antenna radiation pattern after it has been converted to the SCM map format.

Defining an underlay mask is unfortunately not a simple process if no mask parameters or measurement data are available to aid in its construction, as is often the case. Six different methods for generating underlay masks are discussed in [16]:

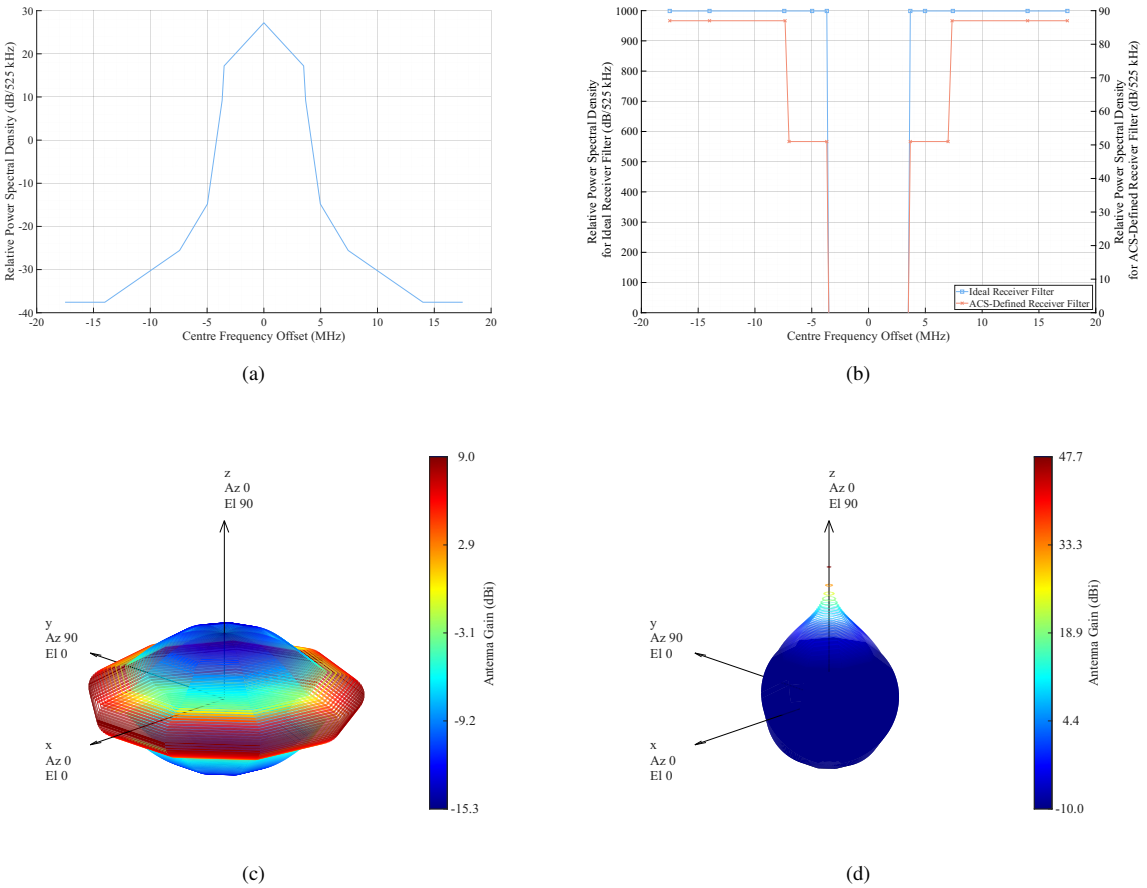


Fig. 5. Masks and 3-D antenna radiation patterns shown for the interfering transmitter ((a) and (c)) and victim receiver ((b) and (d)), respectively. (a) SEM for BWA base station as defined in [17]. (b) FSS earth station underlay mask shown for a perfect filter (blue) and an ACS-defined filter (red). (c) BWA base station monopole antenna radiation pattern. (d) FSS earth station parabolic dish antenna radiation pattern.

1) Assume a perfect receiving filter that has 0 dB attenuation in the Occupied Bandwidth (OBW) and near infinite attenuation elsewhere.
2) An underlay mask that exhibits 0 dB attenuation within the OBW and transitions to set Adjacent Channel Selectivity (ACS) values beyond the OBW.
3) Approximate the underlay mask based on the corresponding spectrum mask of the intended transmitter.
4) Use measured data, such as varying Carrier to Interference Ratio (C/I) thresholds per frequency offset.
5) Employ a Butterworth or Nyquist filter to define a theoretical underlay mask.
6) Follow the method used in ETSI TR 101 854 [20] by combining spectrum mask parameters and theoretical filter models to develop the underlay mask.

This paper's approach follows the first and second methods to show the impact these underlay masks might have on the results. The underlay masks for the FSS earth station are shown in Fig. 5b. Minor frequency offsets have been additionally specified for the subsequent control points within each step function. This allows for power calculations to be performed at the block edges of each mask.

The RBW for the underlay mask is normalised in the same fashion as the spectrum mask using (6) and (7) for normalisation.

Recall the receiver reference power construct from Section IV. The value is determined with (8) as adapted from [4].

$$RP_{RX} = I_T - U_{3\text{dB}} \tag{8}$$

where RP_{RX} is the receiver's reference power, I_T is the maximum permissible long-term interference (20% of the time), and $U_{3\text{dB}}$ is the total power underneath the 3 dB bandwidth of the unscaled underlay mask.

Substituting parameters from Table III in (8) with the calculated I_T value of -151.7 dBW through (3) and the $U_{3\text{dB}}$ value calculated as 11.25 dB via (5), the ideal receiver's reference power is computed as -162.95 dBW. Similarly, for the ACS-defined underlay mask, the $U_{3\text{dB}}$ value is calculated to be 11.27 dB, resulting in a receiver's reference power of -162.97 dBW.

The FSS earth station's power map includes its antenna radiation map, less any insertion losses, which is set to 0 dB in this paper. The antenna radiation map is defined in terms of its main and side lobes in ITU-R S.2196 [21]. Similar to the BWA base station's antenna radiation pattern, the FSS earth station's antenna radiation pattern is originally calculated as antenna gain values in dBi per elevation angle that ranges from -180 to 180 degrees. Each gain value spans the full azimuth range from 0 to 360 degrees. Initially, the convention used for the elevation range does not conform to the one used in the IEEE 1900.5.2 convention and also needs to be converted.

Fig. 5d shows the antenna radiation pattern for the FSS earth station after it has been converted to the SCM format. The parabolic dish antenna pattern of the satellite earth station can be observed.

The placement of the BWA base station and FSS earth station is individually characterised by the location construct, containing latitude, longitude, and altitude. The altitude of the BWA base station and FSS earth station is selected from Table II and Table III, respectively. The latitude and longitude are simplified to a flat-earth grid coordinate system for this problem, as the step sizes of distances are fixed. Additionally, this paper assumes that both services' schedules are continuously active since they are modelled with the same assumption in the ITU-R S.2199 report [5].

VI. Results and Discussion

Refer to the plots in Fig. 6. Four plots are provided detailing the minimum separation distance required for the corresponding FSS earth station antenna elevation pointing angle. Figs. 6a, 6b, and 6c show the results for the co-, first-, and second-adjacent channels, respectively. The fourth plot, Fig. 6d, includes the results from all plots on a single, consolidated axis. The grey areas depict the elevation angles used by the majority of FSS earth stations within South Africa between 27 and 63 degrees, as highlighted by the draft regulations for the Innovation Spectrum recently published by the Independent Communications Authority of South Africa (ICASA) [22]. Table IV summarises the actual minimum separation distances obtained for sampled FSS earth station antenna elevation pointing angles.

The conventional coexistence method found in the ITU-R S.2199 report [5] is first reproduced to ensure that any differences seen with the IEEE 1900.5.2 method can be attributed to the method itself, rather than errors or inconsistencies.

To verify whether our implementation of the conventional coexistence method is correct, a baseline for comparison is established by taking values directly from the original plots in the ITU-R S.2199 report, denoted with the label *ITU S.2199 Copy*. The results obtained by reproducing the conventional method are labelled as *ITU S.2199 Reproduced*. The IEEE 1900.5.2 method is evaluated using two underlay masks: an ideal receiver mask; and an ACS-defined mask, denoted by the labels *ITU S.2199 via SCMs (Ideal RX Mask)*, and *ITU S.2199 via SCMs (ACS-Defined RX Mask)*, respectively.

Refer to Fig. 6a. Good similarity in terms of trend and absolute values is observed on all plots between *ITU S.2199 Copy* and *ITU S.2199 Reproduced*. Visual verification suggests that the results of *ITU S.2199 Copy* are accurately reproduced in *ITU S.2199 Reproduced*, aside from a minimal margin of error.

A. Co-Channel

The *ITU S.2199 via SCMs (Ideal RX Mask)* and *ITU S.2199 via SCMs (ACS-Defined RX Mask)* results show great promise within the co-channel, with separation distances substantially smaller across all antenna elevation pointing angles. The overall trend in the co-channel is extremely similar to the trend found in the *ITU S.2199 Reproduced* results, with smaller separation distances required overall compared to the conventional method. Within this range, employing IEEE 1900.5.2 could be beneficial for enabling spectrum coexistence between PUs and SUs within South Africa since better protection and reduced separation distances are forecasted where necessary. The underlay masks demonstrated identical performance in the co-channel, as the respective 3 dB bandwidths fully overlapped with the spectrum mask within the OBW.

B. Adjacent Channels

The *ITU S.2199 via SCMs (Ideal RX Mask)* and *ITU S.2199 via SCMs (ACS-Defined RX Mask)* results within both first and second adjacent channels performed somewhat poorer in the lower elevation pointing angles than those found in the *ITU S.2199 Reproduced* results. This is expected since the IEEE 1900.5.2 method accounts for the total amount of power under the relevant spectrum mask, which still overlaps with adjacent channels by some margin, even after a guardband is introduced.

Another contributing factor is the directional gain of the FSS earth station at specific elevation pointing angles, since this, combined with differences in antenna heights AGL at smaller separation distances, may utilise the maximum possible directional gain from the antenna radiation pattern. This had a reduced impact on the second adjacent channel compared to the first, as the results were only diminished at the initial elevation pointing angles of 5 and 6 degrees. The *ITU S.2199 via SCMs (Ideal RX Mask)* results exhibit marginally shorter separation distances in the first adjacent channel than those of the *ITU S.2199 via SCMs (ACS-Defined RX Mask)*. This is expected, as the ideal receiver filter perfectly attenuates all out-of-band (OOB) emissions, whereas the ACS-defined mask captures some power leakage from the adjacent channel.

It is important to note that the conventional coexistence method does not account for the continuous spectrum masks. It instead employs a discrete ACLR value to account for interference in adjacent channels as seen in (1) and (2). SCMs allow for evaluating both partially and fully overlapping masks individually, providing a more flexible method to compare desired power with undesired power under each mask's overlapping areas.

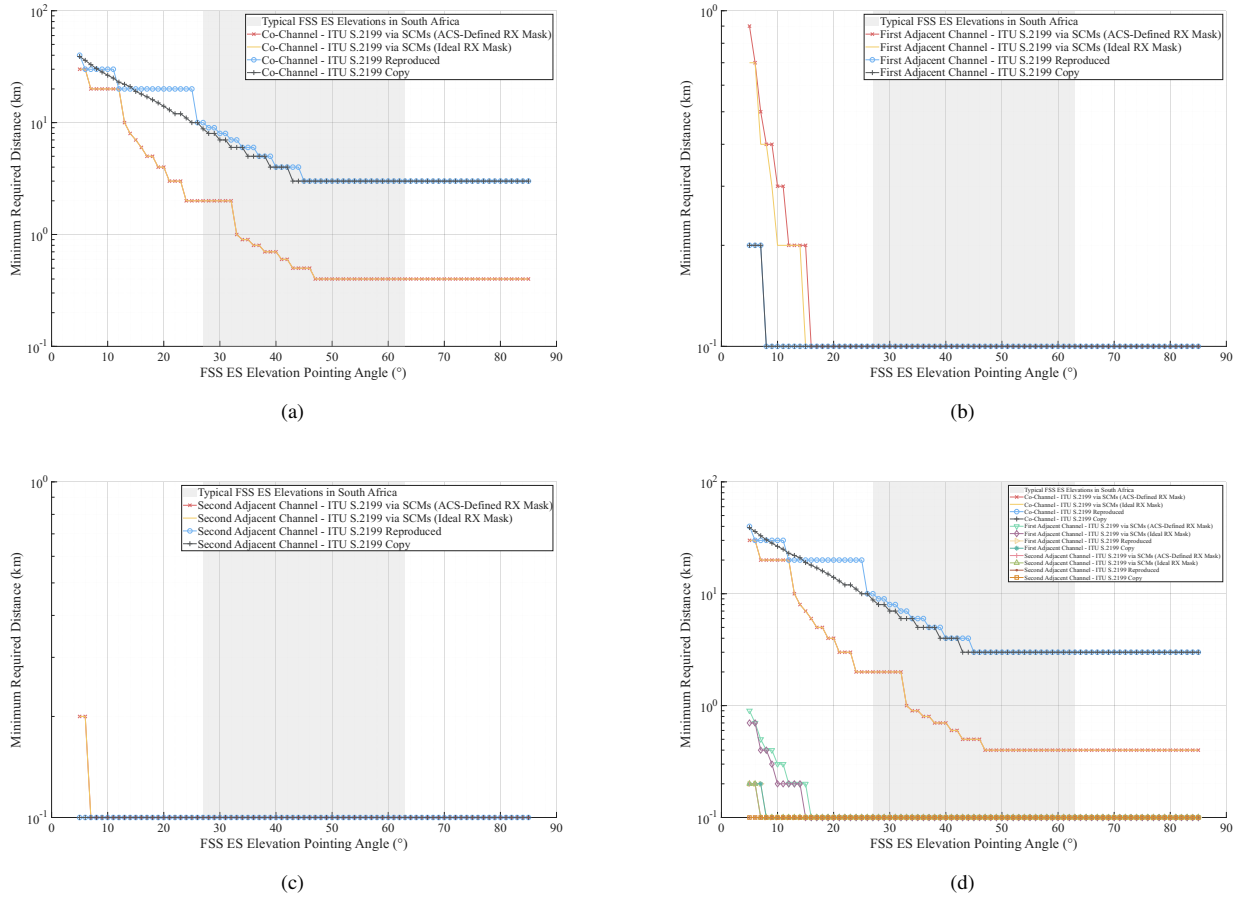


Fig. 6. Minimum separation distances per FSS earth station (ES) elevation pointing angle shown for: (a) the co-channel. (b) The first adjacent channel. (c) The second adjacent channel. (d) All channels evaluated. The grey vertical area between 27° and 63° is the elevation angles used by the majority of FSS earth stations within South Africa.

TABLE IV
MINIMUM SEPARATION DISTANCES BETWEEN BWA BS AND FSS ES PER ELEVATION ANGLE FOR DIFFERENT CHANNELS

Minimum Separation Distance (in km) by Channel	**Spectrum Coexistence Method**	**FSS ES Elevation Pointing Angle (°)**								
		5	15	25	35	45	55	65	75	85
Co-channel minimum separation distances	ITU-R S.2199 Reproduced	40	20	20	6	3	3	3	3	3
	IEEE 1900.5.2 (Ideal RX Mask)	30	7	2	0.9	0.5	0.4	0.4	0.4	0.4
	IEEE 1900.5.2 (ACS-Defined RX Mask)	30	7	2	0.9	0.5	0.4	0.4	0.4	0.4
First adjacent channel minimum separation distances	ITU-R S.2199 Reproduced	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	IEEE 1900.5.2 (Ideal RX Mask)	0.7	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	IEEE 1900.5.2 (ACS-Defined RX Mask)	0.9	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Second adjacent channel minimum separation distances	ITU-R S.2199 Reproduced	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	IEEE 1900.5.2 (Ideal RX Mask)	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	IEEE 1900.5.2 (ACS-Defined RX Mask)	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

Furthermore, it is assumed that most regulatory specifications, including ITU-R S.2199, do not account for any form of a receiver mask to establish performance requirements, since there is no standardised method for constructing such a mask. This introduces numerous issues in constructing a receiver mask due to a lack of standard metrics that characterise the process. A report by the CEPT Electronic Communications Committee (ECC) [23] acknowledges that it is important to look at receivers' performance requirements as well if spectrum compatibility is truly desired.

VII. CONCLUSION

This paper aimed to address the gap in comparing IEEE 1900.5.2 SCMs with a conventional spectrum coexistence method via a comparative analysis. A scenario out of the ITU-R S.2199 report was used as a case study, where an urban BWA base station seeks to operate alongside a co-located FSS earth station.

The IEEE 1900.5.2 and the conventional coexistence method were described in detail. We reproduced the conventional method and compared it to a baseline set by values taken directly from the ITU-R S.2199 report [5] to ensure reliability and clarity when comparing the different methods. The reproduced conventional coexistence method was then compared to the IEEE 1900.5.2 method. Two underlay masks were investigated for the IEEE 1900.5.2 method. The separation distances and the earth station antenna elevation pointing angles at which compatibility is achieved were determined for the co-channel, first adjacent channel and second adjacent channel in each instance.

The findings revealed that the IEEE 1900.5.2 method requires smaller separation distances for service compatibility compared to the conventional coexistence method detailed in [5]. The difference is particularly paramount in the common elevation pointing angle range employed by the majority of South Africa's FSS earth stations. This highlights its significant potential for quickly determining coexistence in FSS-occupied bands by only exchanging minimal amounts of information without the need for detailed operational data or a unified geo-location database.

We demonstrated that employing an ACS-defined underlay mask results in a minor increase in the separation distances compared to an ideal underlay mask in the first adjacent channel. Nonetheless, by integrating continuous underlay and spectrum masks into the calculations, spectrum coexistence could be managed more flexibly via IEEE 1900.5.2, resulting in improved co-channel compatibility between a BWA base station and an FSS earth station. While this study focused on a single interferer, it is also important to assess how aggregate interference from multiple BWA base stations contributes to FSS interference. Future work will apply IEEE 1900.5.2 SCMs to determine additional coexistence opportunities among PUs and SUs in South Africa's proposed Innovation Spectrum [22].

REFERENCES

[1] S. Bhattarai, J.-M. Park, B. Gao, K. Bian, and W. Lehr, "An Overview of Dynamic Spectrum Sharing: Ongoing Initiatives, Challenges, and a Roadmap for Future Research," *IEEE Transactions on Cognitive Communications and Networking*, vol. 2, no. 2, pp. 110–128, 2016.

[2] Y.-C. Liang, *Dynamic Spectrum Management: From Cognitive Radio to Blockchain and Artificial Intelligence*, ser. Signals and Communication Technology. Springer Singapore, 2020. [Online]. Available: http://link.springer.com/10.1007/978-981-15-0776-2

[3] J. A. Stine and S. E. Schmitz, "Model-Based Spectrum Management. Part 1: Modeling and Computation Manual, Version 2.0:," Defense Technical Information Center, Tech. Rep., 2013. [Online]. Available: http://www.dtic.mil/docs/citations/ADA597361

[4] *IEEE Standard for Method for Modeling Spectrum Consumption*, IEEE Std. [Online]. Available: https://ieeexplore-ieee-org.nwulib.idm.oclc.org/document/11060594

[5] Interntional Telecommunication Union (ITU), "Studies on compatibility of broadband wireless access systems and fixed-satellite service networks in the 3 400-4 200 MHz band," Interntional Telecommunication Union (ITU), Tech. Rep. S.2199, November 2010.

[6] Carlos E. Caicedo Bastidas and Arnav Mohan, "Spectrum consumption model builder and analysis tool: A software tool to enhance spectrum sharing," *Analog Integrated Circuits and Signal Processing*, vol. 91, no. 2, pp. 217–226, 2017.

[7] D. Stojadinovic, P. Netalkar, C. E. C. Bastidas, I. Kadota, G. Zussman, I. Seskar, and D. Raychaudhuri, "A Spectrum Consumption Model-based Framework for DSA Experimentation on the COSMOS Testbed," in *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*. ACM, 2022, pp. 77–84.

[8] P. Netalkar, C. E. C. Bastidas, I. Kadota, G. Zussman, I. Seskar, and D. Raychaudhuri. Scalable Dynamic Spectrum Access with IEEE 1900.5.2 Spectrum Consumption Models. [Online]. Available: https://www.techrxiv.org/users/762009/articles/738758-scalable-dynamic-spectrum-access-with-ieee-1900-5-2-spectrum-consumption-models?commit=9a881e75a662632103db00e94048be0212a7bfd8

[9] E. Lagunas, C. G. Tsinos, S. K. Sharma, and S. Chatzinotas, "5G cellular and fixed satellite service spectrum coexistence in C-band," *IEEE Access*, vol. 8, pp. 72 078–72 094, 2020.

[10] B. Lim and M. Vu, "Interference analysis for coexistence of terrestrial networks with satellite services," *IEEE Transactions on Wireless Communications*, vol. 23, no. 4, pp. 3146–3161, 2024.

[11] *IEEE Std 1900.1-2019 (Revision of IEEE Std 1900.1-2008) IEEE Standard for Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management*, IEEE Std., 2019.

[12] *Determination of the interference potential between earth stations of the fixed-satellite service and stations in the fixed service*, Interntional Telecommunication Union (ITU) Std. SF.1006, Rev. 0, April 1993.

[13] *Prediction procedure for the evaluation of microwave interference between stations on the surface of the Earth at frequencies above about 0.7 GHz**, Interntional Telecommunication Union (ITU) Std. P.452, Rev. 13, August 2007.

[14] I. Stevanovic, "Python Implementation of Recommendation ITU-R P.452," Accessed: 18 September 2025. [Online]. Available: https://github.com/eeveetza/Py452

[15] *Sharing studies between International Mobile Telecommunication-Advanced systems and geostationary satellite networks in the fixed-satellite service in the 3 400-4 200 MHz and 4 500-4 800 MHz frequency bands in the WRC study cycle leading to WRC-15*, Interntional Telecommunication Union (ITU) Std. SF.1006, Rev. 0, June 2015.

[16] J. Pahl, *Interference Analysis: Modelling Radio Systems for Spectrum Management*, 1st ed. Wiley, May 2016.

[17] *Fixed Radio Systems; Multipoint Equipment and Antennas; Part 2: Harmonized EN covering the essential requirements of article 3.2 of the RTTE Directive for Digital Multipoint Radio Equipment*, ETSI Std. 302 326-2, Rev. 1.2.2, June 2007.

[18] *Bandwidth Measurement at Monitoring Stations*, Interntional Telecommunication Union (ITU) Std. SM.443, Rev. 4, February 2007.

[19] *Reference radiation patterns of omnidirectional, sectoral and other antennas in point-to-multipoint systems for use in sharing studies in the frequency range from 1 GHz to about 70 GHz*, Interntional Telecommunication Union (ITU) Std. F.1336, Rev. 2, January 2007.

[20] European Telecommunications Standards Institute (ETSI), "Fixed radio systems; point-to-point equipment; derivation of receiver interference parameters useful for planning fixed service point-to-point systems operating different equipment classes and/or capacities," European Telecommunications Standards Institute (ETSI), Tech. Rep. 101 854, January 2005.

[21] Interntional Telecommunication Union (ITU), "Methodology on the modelling of earth station antenna gain in the region of the antenna main-lobe and the transition region between the minimum angle of the reference antenna pattern and the main-lobe," Interntional Telecommunication Union (ITU), Tech. Rep. S.2196, July 2010.

[22] Independent Communications Authority of South Africa, "Draft regulations on dynamic spectrum access and opportunistic spectrum management in the innovation spectrum 3800-4200 MHz and 5925-6425 MHz," in *Government Gazette*. Republic of South Africa: Government Printing Works, 28 March 2025, vol. 717, notice 52415.

[23] Electronic Communications Committee, "Evaluation of receiver parameters and the future role of receiver characteristics in spectrum management, including in sharing and compatibility studies," European Conference of Postal and Telecommunications Administrations (CEPT), Tech. Rep. 310, January 2020.

Best Fit Channel Switching for RF Energy Harvesting Cognitive Radio Networks

Fidel Wasonga, Thomas O.Olwal, Anish M. Kurien
Deparment of Electrical Engineering/F'SATI
Tshwane University of Technology
Pretoria, South Africa
216003500@tut4life.ac.za, OlwalTO@tut.ac.za , kurienAM@tut.ac.za

***Abstract*— In heterogenous cognitive radio network (HCRN) the secondary users (SUs) switch to primary users (PUs) channels opportunistically to improve their radio frequency energy harvesting (RF EH) performance. However, when SUs opportunistically switches to different PUs channels from their current channel, they incur channel switching energy cost. Most of the existing works have only considered the channel switching energy cost in traditional HCRN without RF EH. Additionally, most of the previous studies consider channel switching energy cost in which the SUs is operating in narrow frequency band. The research gap resulting from this consideration is that most of existing work study energy efficient channel switch cost problems without considering RF EH CRN operating in wide frequency band. In order to fill this gap, this paper addresses an energy efficient channel switching optimization problem for wide frequency band CRN with RF energy harvesting technique. The optimization problem is solved by proposing a best fit energy efficient channel switching methods amongst multiple frequency bands. In this approach SUs minimize the average energy cost used in channel switching, while maximizing RF energy harvesting simultaneously. Our numerical results, verified through extensive event-driven simulations and comparative analysis, demonstrate that our algorithm can reduce the average number of channels switching by 40%, 68% and 80% compared to existing algorithms.**

Keywords—Radio frequency, Cognitive radio, primary user, secondary user, energy harvesting

I. INTRODUCTION

Cognitive radio is a promising solution that can help address the scarcity of spectrum resources in highly dynamic radio environment[1].To guarantee the performance of primary users (PUs) in cognitive radio networks (CRNs), secondary users (SUs) need to vacate the channels once the PUs reclaim them. In decentralized CRNs, SUs access the spectrum in an opportunistic fashion by running their own internal dynamic spectrum access (DSA) policy [2]. The spectrum sensing is a pivotal function for CRNs in determining channel state[3]. As a result, in CRNs the SUs are forced to vacate their channels and switch their radios to other unoccupied PU channels more frequently than in traditional networks. On the other hand, CRNs consume significant energy to support CR functionalities such as dynamic spectrum sensing and switching. Nonetheless Radio frequency energy harvesting-based cognitive radio network (RF-CRN) has recently drawn significant interest since it can reuse the licensed spectrum and convert the scavenged RF signals into electricity to realize self-sustained communication systems[4].

II. RELATED WORKS

In [5] the authors considered the problem of throughput optimum channel switching for channel access by a secondary user (SU) in a multichannel cognitive radio network in which there is a delay involved in switching from one channel to another. The lifetime of the energy constrained battery operated SU nodes is critical. Hence, the SUs need to be energy efficient to have a prolonged operation in the network and for more transmission opportunity to increase their throughput. Furthermore, the authors considered SUs operating on wide frequency band in which, a non-negligible energy consumption in channel switching occurs due to the distance between the previous and current frequency. Hence it is notable that, the work in [5] did not consider the energy efficiency during channel switching on wideband frequency.

As result, recent studies in the literature concerning CRNs, have shown a growing interest in energy efficiency during channel switching. In [6], the authors studied energy efficiency of single channel dynamic spectrum access and multi-channel dynamic spectrum access schemes. In multi-channel dynamic spectrum access , compared to sensing or transmission, an SU consumes considerable amounts of energy in channel switching. On the other hand, in a single dynamic spectrum access, the energy consumption due to a channel switching mechanism is eliminated. However, this is at the cost of lower transmission opportunity which may lead to a poor energy efficiency. In addition, the aforementioned work only investigated a single SU case, which is quite simplistic for communication systems. Practical wireless communication networks are inherently multi-users with peculiar benefits and channels.

The authors in [7] studied opportunistic spectrum access problem in multichannel CRNs in which the SU has an energy cost when switching across channels. The authors proposed an energy efficient channel switching policy that minimizes average energy used for a given throughput by modelling an energy efficient switching problem as an average reward Partially Observable Markov Decision Process (POMPD) from which an average energy optimal policy was obtained. In [7], it is notable that the authors only considered channel switching energy consumption on a narrowband frequency. While channel

switching energy consumption might be negligible in narrowband frequency, CRNs are expected to possess operation capability within a wide range of frequency band to experience different channel gains across different frequency channels. In this context CRN can take advantage of the difference in the channel gains when devising a channel exploitation strategy for the SUs, thereby plausibly increasing resource utilization. Hence, channel switching energy consumption on wide frequency band should be taken into account.

An energy energy-efficient channel switching scheme was presented in [8]. Unlike existing schemes where the SUs adopt the same channel switching strategies regardless of which channel they currently stay at .In addition the scheme in [8] allows SUs to adapt their channel switching via reinforcement learning problem which permits SUs to learn channel switching schemes by interacting with the environment. Despite their inherent energy efficient design, these work did not consider RF EH aspect which can enhance energy efficiency and provide extend network lifetime [8].

In [9], the authors proposed a joint frequency assignment and scheduling problem for multihop industrial CRNs with a single transceiver by considering varying amounts of energy consumption that occurs in CRN when switching to different frequency bands. In the problem the authors formulated a linear optimization problem which minimizes the energy cost related to channel switching. However, the authors in [9] considering minimizing the energy cost of channel switching in wide frequency band . and considered the case where a CR device has a single transceiver instead of multiple transceivers which could potentially decrease the energy consumption due to the channel switching since less switching needs to be performed. In addition, having a single transceiver increases channel switching cost since the single transceiver has to give service to each wireless link of CRN.

Overall we note that, although there have been prior studies in channel switching considering energy efficiency in CRN [7] [8]. To the best of the authors knowledge, no studies in the literature have considered the impact of channel switching overhead in terms of energy efficiency in CRNs incorporating RF energy harvesting technique while operating in wideband frequency. Motivated by this, this article aims to minimize the energy cost of the SU's in channel switching energy, while at the same time maximizing RF energy harvesting.

.

To summarize, the important distinctions between our paper and existing studies are as follows:

- We present a model that captures opportunistic RF energy harvesting in a multi-user multichannel CRNs while, the existing methods consider CRN without RF energy harvesting operating in narrow frequency band
- The primary goal of this work is to minimize SUs in CRN channel switching energy cost while at the same time maximizing RF energy harvesting. By proposing best fit channel switching algorithm for radio frequency (RF) energy harvesting, SUs in CRN minimize the energy cost incurred during channel switching while operating in wide frequency band. However, the primary goal of the existing studies is to minimize channel switching energy cost in traditional SUs without RF energy harvesting operating in narrow frequency bands.
- Simulation results demonstrate Best Fit Channel Switching algorithm has considerable improvement in channel switching overhead efficiency compared to POMPD 58%, RLA 80 % and SCAFAS 75 % algorithm

III. SYSTEM MODEL

In this section, we begin by introducing a cognitive radio network comprising of static primary network and an energy-harvesting secondary network. We consider a heterogeneous CRN with two types of users are present in the same communication range and they are the licensed PUs annotated as {Y1,Y2…. Yi} and unlicensed users SUs equipped with RF energy harvesting capability annotated as {X1, X2…Xi} as shown in Fig 1. It is assumed that the PUs are distributed within the coverage area of heterogenous CRN The PUs as shown in Fig. 1 are centered in a guard zone and a harvesting zone that is inside the guard zone. Hence, when the SU is in harvesting mode and if it is within the harvesting zone of an active PU it is guaranteed to harvest sufficient RF energy to charge it to the threshold. On the other hand, if an SU is outside any harvesting zone the power received will be too small to active RF energy harvesting. For instance, in Fig 1 an SU, which lies in the harvesting zone of a PU on channel 1, lies outside the guard zones of all active PUs on channel 2. Generally, the operating power of the RF energy harvesting SUs is much higher and as a result the RF energy harvesting zones are smaller.

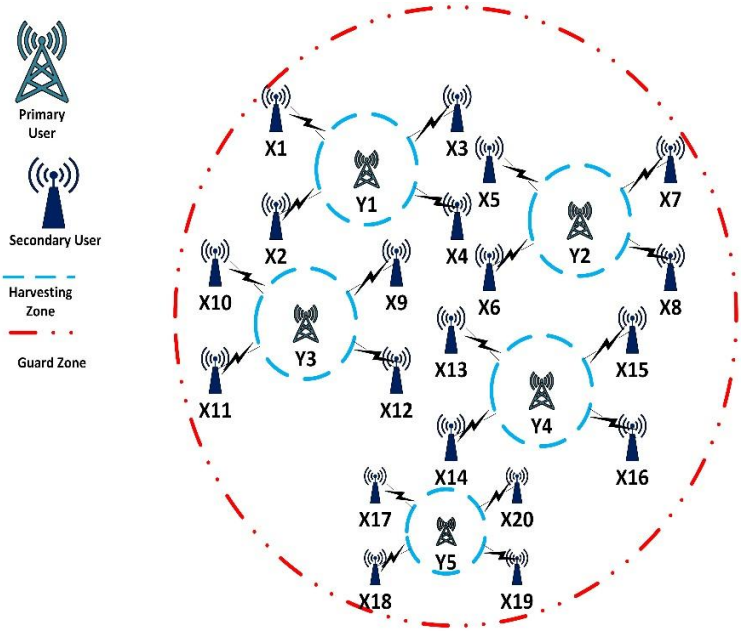


Figure 1 Heterogenous cognitive radio network [10].

A. Primary users' channel usage model

A dynamic spectrum access scenario is considered with multiple PU channels. The PU channel model we use is the two-state OFF-ON model and it is assumed that the occupancy of each channel by the PU evolves independently according to a homogeneous continuous-time Markov chain. This channel model has been widely used in the literature[11]. For intuitive understanding on system performance through closed form analysis OFF-ON period lengths are assumed which are exponentially distributed with the mean values, τ_n and μ_n. Further, in the OFF-ON exponential modeling, based on the PU activity vector constructed over a defined time, distributions of OFF and ON time intervals are approximated to two exponential distributions. In addition, the assumption is made that the distributions of ON/OFF or idle/busy time are independent and not the same. During each time slot, the PUs channels can be idle or busy. It is assumed that the SU's can access any one of all the available PU channels. If the PU channel is sensed to be idle, the SU will use it and sense it continuously until the PU is present. Once the PU is present in the channel, the SU will switch to the next available channel. Also, it is assumed that, when the SU is using an idle channel, it will sense continuously that channel and stop harvesting it when the PU becomes ON. We consider the SU with RF energy harvesting capability. The SU can harvest RF energy from the PUs transmission when the PU channel is busy to extend their battery life and cannot while the PU channel is idle.

The OFF/ON channel states are represented in terms of discrete time Markov chain as [12]:

$$P_{OFF\to ON} \triangleq_{P0,1} = \int_0^{T_s} \frac{1}{\tau_n} e^{-b/\tau_n} db = 1 - e^{-T_s/\tau_n} \quad (1)$$

$$P_{ON\to OFF} \triangleq_{P1,0} = \int_0^{T_s} \frac{1}{\mu_n} e^{-b/\mu_n} db = 1 - e^{-T_s/\mu_n}, \quad (2)$$

where τ_n and μ_n are OFF and ON exponentially distributed mean values and T_s is the slot duration. The slot duration is portioned into four non-overlapping portions: channel switching time T_{cs} , learning time T_{ln} , control messaging time T_{cm} , data transmission $T_s - T_{cm} - T_{ln} - T_{cs}$. T_{cs} is the total time required for completing the channel switch which is given as follows:

$$T_{cs} = t_{cs}|f - f^i|, \quad (3)$$

where f^i is the current PU channel transmission frequency, f is the assigned PU channel frequency. If a SU does not switch to a PU channel, the channel switching time becomes zero $T_{cs} = 0$. Let f_n^0 denote the frequency of the PU channel to which the antenna of the SU is tuned at the beginning of the quiet period. The T_{cs} time depends on Phase-Locked Loop (PLL), bandwidth, settling time, and the step size in frequency. During, the circuitry is active but not performing RF energy harvesting efficiently. This transient energy consumption accumulates as the channel switch energy (E_{CS}).

B. Energy consumption modelling

The energy consumed for configuring the hardware from the current transmission frequency f^i to the assigned transmission frequency f is denoted by E_{CS} , and it depends on the cognitive radio hardware. $E_{CS} = 0$ if the PUs channel frequencies are the same; however, it is symmetric and nonzero for different frequencies. The total energy consumption due to channel switching (E_{CS}) for one switch is as follows:

$$(E_{CS}) = P_{CS}T_{CS}^{i,f} = P_{CS}.T_{cs} = t_{cs}|f - f^i| \, Joules \quad (4)$$

where P_{CS} is the power dissipation for switching and $T_{CS}^{i,f}$ is the switching time for one frequency jump. The switching time typically increases with frequency step size.

Total energy for sensing set C_n by SU_n is as follows:

$$E_{CS} = P_{CS}.T_{cs}.\left(|f_0^n - f_1^n| + \sum_{k=1}^{|C_n|}|f_k^n - f_{k+1}^n|\right) \quad (5)$$

where P_{CS} is the power consumption during switching T_{cs} is the time required for each switch, f_0^n is the unitial tuned frequency of SU_n, C_n is the ordered set of channels SU_n will sense.

The total switching energy for all SUs is calculated as follows from equation (5)[13]:

$$E_{CS} = E_{CS}^n = P_{CS}.T_{cs}.\sum_{n=1}^{N}\left(|f - f^i| + \sum_{k=1}^{|C_n|-1}|f_k^n - f_{k+1}^n|\right) \quad (6)$$

$$E_{CS} = P_{CS}.T_{cs}.\left(|f - f^i| + \sum_{k=1}^{|C_n|-1}|f_k^n - f_{k+1}^n|\right) \quad (7)$$

C. Energy harvesting model

There are different energy harvesting models that are provided in the literature e.g. non-linear energy harvesting model, linear energy harvesting model and piece-wise linear energy harvesting model. In simultaneous energy harvesting and cooperative spectrum sensing model, we assume that SU's does not have a fixed power supply and adopts one energy harvesting device, which can harvest RF energy opportunistically from the PUs channels transmission to replenish its battery. The harvested energy from the PUs channels transmission is stored in the battery (e.g. super capacitor) on each SU with the maximum battery capacity $\mathcal{M}E_h$, where $\mathcal{M}$ is a positive integer and E_h denotes the amount of energy that could be harvested by each SU during one slot. The energy harvested by the SU on a given channel overtime slot T is expressed as follows:

$$E_h(t) = \eta.\sum_{n=1}^{N} P_r^{(n)}(t) \cdot T_h^{(n)} \quad (8)$$

where $E_h(t)$ is the total harvested energy in slot T, η is energy harvesting efficiency, $P_r^{(n)}$-is the received RF power in subband

n at time, $T_h^{(n)}$ is the fraction of time spent harvesting RF energy from subband n.
The energy storage battery dynamics are formulated as follows:

$$B(t+1) = \min\left[B_{max}, B(t) - E_{cs}(t) + E_h(t)\right] \quad (9)$$

where B_{max} is the battery capacity, $B(t)$ is the stored energy at slot T, $E_{cs}(t)$ energy consumed in channel switching in wideband frequency, and $E_h(t)$ is the harvested energy during the slot.

In RF energy harvesting CRN, SUs should carefully evaluate whether to switch to a new channel by comparing: the energy harvested from the new channel against the energy cost of switching to that channel. This is important to RF energy harvesting CRN with variable energy harvesting rate or limited energy storage which is given as follows:

$$E_h > (E_{CS}) = P_{CS}.T_{cs}\ \ Joules \quad (10)$$

Due to the battery size having an important impact on network performance, and the energy harvesting capacities, we assume that the battery size is enough for the energy harvested from the PUs channels transmission to support the switching between spectrum sensing, data transmission and energy harvesting.

IV. Channel Switching Problem Formulation

In this study, we formulate an energy efficient channel switching optimization problem which minimizes the average energy cost used in channel switching by the SUs while at the same time maximizing RF energy harvesting
At each time slot T SUs should switch from the current PU channel f_{cur} to a new PU channel f^i only if the net energy gain is positive to avoid wasting harvested RF energy on unnecessary channel switching. Therefore, SUs should only switch to a new channel if the expected energy gain from harvesting on the new channel outweighs the energy cost of switching. That is only if the energy gain from the new channel not only pays for the switch but also provides enough surplus harvested RF energy for battery storage. This is provided by the first principle of the switching rule which explicitly accounts for battery capacity and sensing cost. Switching from current channel i (at t-1) to candidate channel j (at t) as follows:

$$E_{h,new} - E_{h,current} > E_{CS}(i,j) + E_{sense}(j) \quad (11)$$

where $E_{h,new}$ is the new RF energy harvested, $E_{h,current}$ is the current harvested energy, E_{CS} is the channel switching energy cost, and $E_{sensing}$ is the candidate channel j sensing energy. At each slot the SUs computes the immediate expected net energy gain from switching to every other channel which is given by equation (17). If the best positive net gain exceeds a margin, switch to the PU channel with maximum gain otherwise stay. The equation of current channel harvested energy is represented as follows:

$$E_{h,current}^{(i)} = \eta.\sum_{n=1}^{N} P_i^{RF}(t)\cdot T_h^{(n)} \quad (12)$$

where i is the current channel index, η RF-DC conversion efficiency, $\mathcal{T}$ slot duration, $P_i^{RF}(t)$ received RF power on channel i at slot t.

If you switch, your battery after the slot while ignoring other drains changes by:

$$\triangle B_{switch} = E_{h,new} - E_{CS}(i,j) + E_{sense}(j) \quad (13)$$

If the SUs stays, the battery change is:

$$\triangle B_{stay} = E_{h,current} \quad (14)$$

Otherwise, if the switching is beneficial to the SUs:

$$\triangle B_{switch} > \triangle B_{stay}$$

$$E_{h,new} - E_{sense} - E_{CS} > E_{h,current}$$

If the SUs stays, the battery change is:

$$\triangle B_{stay} = E_{h,current} - E_{sense}(j) + E_{CS}(i,j) \quad (15)$$

Channel-aware switching decision is based on switch if the net energy gain is positive which is given as follows:

$$E_h^{(j)} > E_h^{(i)}(t) + E_{CS}(i,j) \quad (16)$$

Channel-aware switching equation (x) ensures the SUs only switch if the expected harvested energy advantage on the new channel outweighs the combined sensing and channel switching energy cost.

For the threshold-based rule, let $\triangle E = E_{h,new} - E_{h,current}$

Switch if:

$$\triangle E > E_{CS}(i,j) + \in \quad (17)$$

where $\in$ is a safety threshold to prevent frequent switching. The factors considered in the decision rule are the channel quality (SNR), battery level and PU activity.

We define the system state at time as:

$$s_t = \left(E_t, C_t, \{P_t^i\}_{i=1}^N, \{z_t^i\}_{i=1}^N\right) \quad (18)$$

where E_t is the battery energy level, C_t current channel occupied by the SU, P_t^i PU occupancy state on channel, z_t^i channel quality (SNR) of channel
So, if the SU was on channel i and switches to channel j :

$$E_{CS}(i,j) = \begin{cases} 0, & i=j \\ E_0 + k\Delta f(i,j) & i\neq j \end{cases} \quad (19)$$

where E_0 is the fixed overhead (energy to switch), k is the scaling factor (extra cost for wideband frequency jumps), Δf is the frequency separation between channels. In wideband RF EH CRN, E_{CS} is especially critical because exploring many subs band naively can waste more energy than is harvested. Naively switching across many sub-band multiple channel switch events consequently leading to RF EH CRN energy draining quickly. When k is large switching, it is catastrophic due to the expected number of candidate switches per unit time grows.

A. PROPOSED SOLUTION

In this section, we propose Best Fit Channel switching algorithm. Our algorithm assigns SUs to channels sequentially and updates the channel selection sequence to obtain the least channel switching energy while at the same time maximizing RF energy harvesting. For finding the best PUs channels for RF energy harvesting, an SU first tunes to the PUs channels with the maximum or minimum frequency depending on which one is closest to its initial frequency (f_{cur}). Although the channel switching problem is addressed in various frameworks, the novelty of the problem is that the formulation translates into a joint design problem.

We now present the objective function of our algorithm with an optimization model that minimizes average switching energy while maximizing average harvested energy. Let T be horizon, $w \in [0,1]$ trade off switching cost and harvested energy. If $w = 0$ it maximizes harvested energy only and if $w = 1$ the objective reduces to minimizing the average switching energy only. The optimization model can be written as:

$$(\mathrm{P}) \min_{x_i(t),s_i(t),u_i(t),z_{ij}} \frac{1}{T}\sum_{t=1}^{T}[w(E_{CS}(t) - (1-w)\sum_i E_h(t)] \quad (20)$$

$$\text{s.t } \sum_{i=1}^{N} x_i(t) \le 1,\ x_i(t) \in \{0,1\}, u_i(t) \in \{0,1\}, s_i(t) \in \{0,1\},$$
$$z_{i,j}(t) \ge x_i(t-1) + x_j(t) - 1, \sum_{i,j} z_{i,j}(t) \le 1$$

In general, it can be challenging to obtain the optimal channel switching solution by directly solving the optimization problem (21). Therefore, our aim is to derive an equivalent formulation of problem (21), which leads to a low-complexity solution for the optimal channel switch strategy. Therefore, our aim is to derive an equivalent formulation of the problem in (21) which leads to a low complexity solution for optimal channel switching strategy. In the following proposition this alternative optimization problem is presented:

$$\min_{x,z} \frac{1}{T}\sum_{1}^{T}[(1-w)]\sum_{i\epsilon C}\sum_{j\epsilon C} Z_{i\to j}(t)E_{CS}^{i\to j}(t) - w\sum_{j\epsilon C} x_j(t)E_{h,j}(t) \quad (21)$$

s.t $\sum_{j\in C} x_j(t) = 1,\ x_j(t) \in \{0,1\}\ \forall j,t,$ and linking constraints.
$z_{i\to j}(t) \le x_i(t-1)$, $z_{i\to j}(t) \le x_j(t)$, $z_{i\to j}(t) \ge x_i(t-1) + x_j(t) - 1$

The objective minimizes average net cost which is the weighted switching cost minus the weighted harvested energy. Because $E_{h,j}(t)$ enters with a negative sign, maximizing harvested energy is equivalent to minimizing the objective. Optimization problem in (20) is not only more convenient than the one in (21) which involves indicator functions but also leads to simpler formulations of the optimal channel problem. To conclude, the proposition provides a scaled and more compact version of the optimization problem (20). The optimization problem in (21) can further be defined with a target RF harvested energy $E_{h,ref}$ .By using $E_{h,ref}$, the nonnegative -form minimization objective can be expressed as follows:

$$\min \frac{1}{T}\sum_{t=1}^{T}\left[[(1-w)\sum_{i,j} z_{i\to j}(t)E_{CS}^{i\to j}(t) + w(E_{h,ref} - \sum_j x_j(t)E_{h,j}(t)\right] \quad (22)$$

where $E_{h,ref}$ is chosen so $E_{h,ref} - \sum_j x_j(t)E_{h,j}(t) \ge 0$ example $E_{h,ref} = max_{j,t}\, E_{h,j}(t)$. Switch from the current channel c at $t-1$ to candidate j at t only if the change reduces the objective to as follows:

$$(1-w)E_{CS}^{C\to J}(t) - w\left(E_{h,j}(t) - E_{h,c}(t)\right) < 0 \quad (23)$$

Equivalent equation (23) can be expressed as follows:

$$E_{h,j}(t) - E_{h,c}(t) > \frac{1-w}{w}E_{CS}^{C\to j}(t)\ \ (w \in (0,1) \quad (24)$$

Hence when $w = 0$ the SU no channel switching occurs on RF energy harvesting from the selected PU channel. On the other hand, when $w = 1$ the SU ignores switching cost and always picks the PU channel with maximum RF energy harvesting. This ensures that switching is energy efficient and extends the operation lifetime of energy-constrained CRN.

Best Fit Channel Switching Algorithm

```
Inputs:
  T       # horizon
  C       # number of channels
  w ∈ [0,1] # trade-off weight
  E_{h,j}(t) # harvested energy for channel j at time t
  # switching cost from channel i to channel j
  C_o     # initial channel

Procedure Switching Policy:
  c = c0  # current channel at time 0
  for t = 1 to T do
    best channel = c
    best delta  = 0  # improvement (negative cost change)

    for j in 1..C do
      # compute incremental cost if switching from c to j
      delta cost =(1 − w)E_CS^{C→J}(t) − w(E_{h,j}(t) − E_{h,c}(t))

      if delta_cost < best delta then
        best delta  = delta cost
        best channel = j
      end if
    end for

    # update decision
    if best channel != c then
      c = best channel  # switch
    else
      c = c          # stay
    end if

    # (Optional) record chosen channel and harvested energy
  end for
```

V. SIMULATIONS AND RESULTS

In this section the performance of the RF energy harvesting cognitive radio network is evaluated. The channel assignment is centrally implemented as the SU's base stations based on outdated channel state information (CSI). We consider RF energy harvesting cognitive radio network deployment region is considered which is distributed in an area of $500 \times 500\ m^2$ with the formation of a 3×3 grid. In order that one hop neighborhood placed on the grid and not in the diagonal can directly communicate with each other. The PU's and SUs are randomly distributed in two-dimensional Euclidean space[14]. Furthermore, the operation range of each PU is set to 170m radius. If two SU's are within the operation range of each other, then there is an edge connecting the two vertices corresponding to these SUs. An SU operating within the range of the PU cannot use the channel used by the PU in that time slot. For simplicity, each SU is assumed to have multiple transceivers and thus can connect to at most to two PUs channels. In addition, since it is a densely populated urban region is assumed, the number of mobile SUs varies from $10{\sim}40$. Moreover, the SUs follows the random waypoint mobility model. This assignment is fixed and known to all users of the network. The Monté Carlo Simulations model is adopted to obtain average results of over 500 different iterations. The results were acquired using MATLAB. Below shows a list of parameters used along with their values during the simulations are listed in the table below without loss of generality.

Table 1 Simulation Parameters

Parameter	Value
Transmission Range	200 m
Simulation Area	500m X 500m
Number of SUs	[10,40]
Number of PUs	[10,40]
Channel switching power	1000 mW
Mobility	Random Mobility
Energy harvesting rate	[-20dBm, -15dBm]
Maximum velocity of PUs	10ms
Simulation total time	25-125 s
Frequency switching probability of PUs	0.1
Average number of channels switching	F/10
Time required for channel searching, selecting and switching	5s

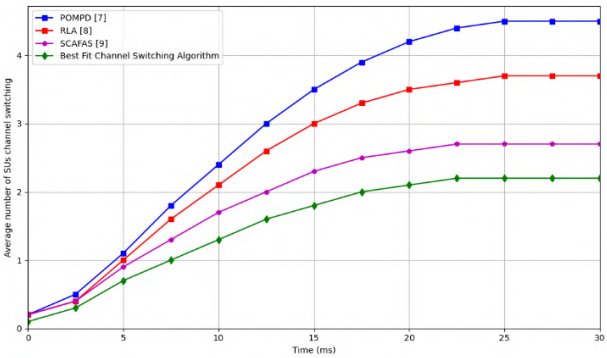


Figure 2 Time versus Average number of channels switching

Figure 2 provides a comparison between time and the average number of channels switching. The maximal optimal value of time duration from 0 to T. From figure 2 it can be observed that the number of channel switches for the proposed method is lower (approximately 40% and 85% compared to traditional channel unaware channel switching scheme. The proposed channel switching scheme decreases the channel switching numbers. This is because the proposed r algorithm limits the number of unnecessary SUs channel switches.

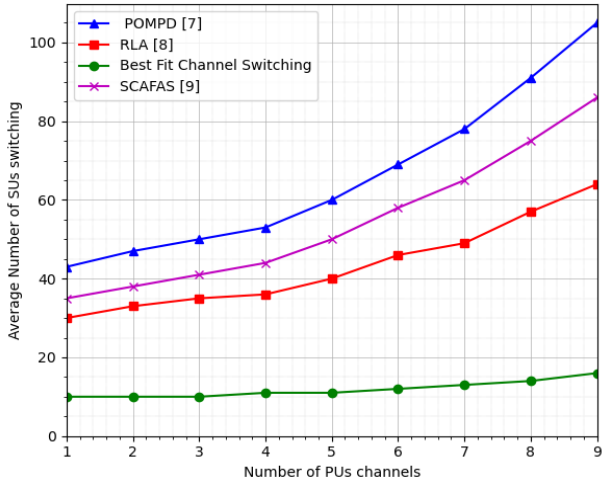


Figure 3 Number of PUs Channels VS Average Number of SUs Switching

In Figure 3, as PUs increases, the channels that are available to the SUs increase and hence, the SU's are obliged to switch channels more frequently. However, best fit channel switching method tries to limit the number of unnecessary SU's channel switching. This figure demonstrates the strength of best fit channel switching method since it outperforms SCAFAS, POMPD and RLA algorithm.

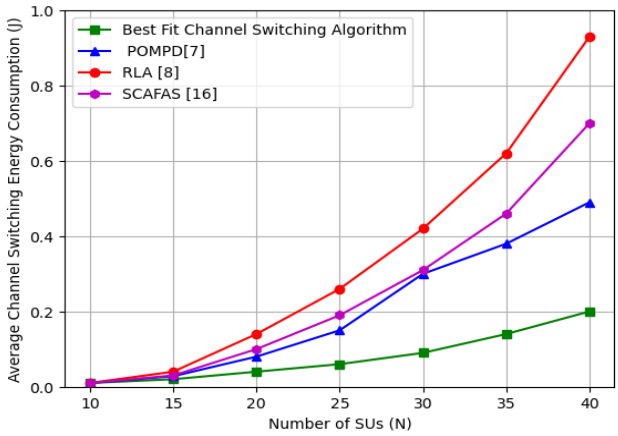


Figure 4 Number of SUs vs Average Channel Switching Energy Consumption.

Figure 3 displays energy consumption due to channel switching with different numbers of SU's. One can observe that energy consumption increases with increasing number of SU's. Since the PUs channels that are available decreases as the number of SU's increases, SU's are forced to switch PUs channels more often. This situation naturally leads to higher energy consumption due to SU's more frequent obligation to switching PUs channels. The results demonstrate that the proposed novel channel switching algorithm yields better performance in comparison with SCAFAS, RLA, and POMPD algorithms.

VI. Conclusion

This paper considered the overhead of channel switching in RF EH CRNs in terms of energy efficiency. By proposing best fit channel switching algorithm. In the SUs you can select multiple PUs channels that are in ON state for RF EH. However, in contrast with existing approaches, the proposed algorithm jointly accounts for the PU traffic characteristics as well as the instantaneous channel state information (CSI) over a particular PU channel that could possibly yield higher RF EH. The results obtained via simulations indicate that our proposed solution outperforms the benchmark scenarios. Future work on this study will consider to the analysis of the impact of channel switching latency/delay and competition among multiple SUs to in selecting the best PU channels for RF energy harvesting on throughput.

ACKNOWLEDGMENT

The authors acknowledge Telkom CoE, Nokia, TUT/FSATI for the financial support in this project. Views and suggestions expressed in this paper emanated from the writers alone.

.

References

[1] Z. Xu, I. Petrunin, and A. Tsourdos, "Identification of Communication Signals Using Learning Approaches for Cognitive Radio Applications," *IEEE Access*, vol. 8, pp. 128930–128941, 2020, doi: 10.1109/ACCESS.2020.3009181.

[2] M. C. Hlophe and B. T. Maharaj, "AI Meets CRNs: A Prospective Review on the Application of Deep Architectures in Spectrum Management," 2021, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2021.3104099.

[3] K. Srisomboon, L. Pipanmekaporn, A. Prayote, and W. Lee, "Adaptive Cooperative Quality Weight Spectrum Sensing for Mitigating Byzantine Attacks in Cognitive Radio," *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3567339.

[4] H. Xiao *et al.*, "Energy-Efficient Resource Allocation in Radio-Frequency-Powered Cognitive Radio Network for Connected Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5426–5436, Aug. 2021, doi: 10.1109/TITS.2020.3026746.

[5] M. Santhoshkumar, D. J. Muttath, and K. Premkumar, "Throughput Optimal Opportunistic Channel Switching in Cognitive Radio Networks," *IEEE Wireless Communications Letters*, vol. 10, no. 9, pp. 2046–2050, Sep. 2021, doi: 10.1109/LWC.2021.3091635.

[6] S. Agarwal and S. De, "Impact of channel switching in energy constrained cognitive radio networks," *IEEE Communications Letters*, vol. 19, no. 6, pp. 977–980, Jun. 2015, doi: 10.1109/LCOMM.2015.2416334.

[7] M. Santhoshkumar and K. Premkumar, "Energy-Efficient Opportunistic Spectrum Access in Multichannel Cognitive Radio Networks," *IEEE Networking Letters*, vol. 5, no. 1, pp. 1–5, Nov. 2022, doi: 10.1109/lnet.2022.3221795.

[8] H. Ding, X. Li, Y. Ma, and Y. Fang, "Energy-Efficient Channel Switching in Cognitive Radio Networks: A Reinforcement Learning Approach," *IEEE Trans Veh Technol*, vol. 69, no. 10, pp. 12359–12362, 2020, doi: 10.1109/TVT.2020.3006471.

[9] S. Demirci and D. Gozupek, "Switching Cost-Aware Joint Frequency Assignment and Scheduling for Industrial Cognitive Radio Networks," *IEEE Trans Industr Inform*, vol. 16, no. 7, pp. 4365–4377, Jul. 2020, doi: 10.1109/TII.2019.2950563.

[10] R. Kishore, S. Gurugopinath, S. Muhaidat, P. C. Sofotasios, O. A. Dobre, and N. Al-Dhahir, "Sensing-throughput tradeoff for superior selective reporting-based spectrum sensing in energy harvesting HCRNs," in *IEEE Transactions on Cognitive Communications and Networking*, Institute of Electrical and Electronics Engineers Inc., Jun. 2019, pp. 330–341. doi: 10.1109/TCCN.2019.2906915.

[11] Y. Saleem and M. H. Rehmani, “Primary radio user activity models for cognitive radio networks: A survey,” 2014, *Academic Press*. doi: 10.1016/j.jnca.2014.04.001.

[12] P. Mukherjee and S. De, “A System State Aware Switched-Multichannel Protocol for Energy Harvesting CRNs,” *IEEE Trans Cogn Commun Netw*, vol. 6, no. 2, pp. 669–682, 2020, doi: 10.1109/TCCN.2019.2942924.

[13] S. Eryigit, S. Bayhan, and T. Tugcu, “Channel Switching Cost Aware and Energy-Efficient Cooperative Sensing Scheduling for Cognitive Radio Networks.”

[14] S. Aslam, W. Ejaz, and M. Ibnkahla, “Energy and spectral efficient cognitive radio sensor networks for internet of things,” *IEEE Internet Things J*, vol. 5, no. 4, pp. 3220–3233, Aug. 2018, doi: 10.1109/JIOT.2018.2837354.

Assessing the Performance of Wi-Fi Student Networks at the University of Zululand's KwaDlangezwa Campus

Dumile Gumede
Department of Computer Science
University of Zululand
KwaDlangezwa, South Africa
201914265@stu.unizulu.ac.za

Alfredo Terzoli
Department of Computer Science
University of Zululand
KwaDlangezwa, South Africa
TerzoliA@unizulu.ac.za

Nombuso Sibeko
Department of Computer Science
University of Zululand
KwaDlangezwa, South Africa
Sibekon@unizulu.ac.za

***Abstract*—This study evaluated the performance and coverage of the UNIZULU-STUDENTS Wi-Fi network at the University of Zululand's KwaDlangezwa campus by integrating student feedback with technical measurements. A mixed-methods approach was used, combining an online survey, site surveys, and performance testing to assess coverage, signal strength, and user satisfaction. Survey results revealed dissatisfaction, with students identifying locations such as the library, lecture theatres, residences, and sports grounds as persistent problem areas. Over 80% responses rated reliable, high-speed Wi-Fi as important for academic and extracurricular activities. Signal mapping with Vistumbler confirmed weak coverage in these locations, with RSSI values ranging from –100 dBm to –64 dBm. Performance tests with iPerf3 further highlighted instability: TCP throughput in the library averaged only 439 Kbps, while UDP tests recorded 90 ms jitter and 68% packet loss. By contrast, controlled tests in the University's Computer Science Honors Laboratory achieved stable throughput of ∼300 Mbps, showing the backbone network was capable but poorly extended to student spaces. The findings point to poor access point placement, congestion, and interference as the main limitations. Targeted improvements are essential to ensure consistent, high-quality campus connectivity that can support digital learning and student success.**

***Index Terms*—Wi-Fi, Signal Strength, Network Performance, User Satisfaction, KwaDlangezwa**

I. Introduction

Wi-Fi networks are now at the centre of modern digital connectivity, supporting communication, collaboration, and productivity across homes, offices, campuses, and public spaces. They work by transmitting data between devices and access points via radio frequencies, giving users the freedom to stay connected while on the move. This capability enables a wide range of applications, including online learning, digital collaboration, entertainment, and everyday communication. [1] and [2] highlighted Wi-Fi's convenience and multi-device support, while [3] emphasized its role in maintaining mobility. Because of these advantages, Wi-Fi has become an indispensable part of daily life. In Higher Education, stable and reliable Wi-Fi access has become essential for academic success, providing students with access to e-learning platforms, enabling communication between lecturers and students, and supporting real-time peer collaboration [4].

Despite this importance, Wi-Fi continues to suffer from persistent reliability challenges such as inconsistent speeds, unstable connections and frequent service dropouts, all of which negatively affect user satisfaction [5]. These problems have intensified with the rapid growth of mobile Internet use and the increasing demand for Quality of Service (QoS), particularly in high-density environments like University campuses. Although Wi-Fi and cellular networks dominate global wireless connectivity because of their low cost, speed, and scalability [6], maintaining consistent performance remains a critical issue.

At the University of Zululand's KwaDlangezwa campus, the institution's main campus, these challenges were especially noticeable. The continuous growth in student and staff population has placed increasing strain on the infrastructure, resulting in poor performance of the UNIZULU-STUDENTS Wi-Fi network. Students often report slow Internet speeds, difficulties accessing e-learning platforms such as Moodle, and unreliable connections in key areas including the library, lecture theatres, residences, and sports grounds [7]; [8]; [9]. Such limitations not only hinder academic activities but also restrict access to essential communication and collaboration tools, highlighting the urgent need for improvements to the network.

To address these issues, this study systematically assessed the performance of the KwaDlangezwa campus Wi-Fi network. A mixed-methods approach was employed, combining an online survey of student experiences with technical data from Wi-Fi coverage mapping (Vistumbler) and performance testing (iPerf). The study evaluated signal strength, coverage and overall connectivity quality to identify bottlenecks. By integrating both student feedback and objective performance measurements, the research produced actionable insights to enhance the reliability and efficiency of Wi-Fi services at the University of Zululand, thereby supporting improved academic performance and student success.

II. Related Work

Numerous studies have examined the effectiveness and challenges of campus Wi-Fi networks. In South Africa, [4] investigated Wi-Fi usage in universities, highlighting its role in

flexible learning but also reporting persistent challenges such as interference, limited coverage, security risks, and poorly configured access points. Their findings indicated that many hotspots were affected by overlapping channels, which often caused congestion, and in some campus locations the hotspots consistently recorded signal strength below –70 dBm, indicating clear coverage gaps.

In a related study, [10] studied University campuses in Ghana and identified recurring issues of low bandwidth, congestion, and interference. Using the inSSIDer software tool, they observed throughput levels dropping below 2 Mbps during peak hours, with certain locations experiencing signal strength weaker than –80 dBm. To address these deficiencies, they proposed a dual-band setup (2.4/5 GHz) combined with high-gain antennas to improve overall performance.

Student satisfaction was also closely tied to Wi-Fi quality. Surveying 260 Malaysian students, [11] found that reliability, responsiveness, and coverage were the strongest predictors of satisfaction. Their regression analysis revealed that responsiveness had the greatest impact ($\beta = 0.529$), highlighting the importance of not only infrastructure improvements but also timely technical support in shaping user experiences.

Coverage mapping has been another focus of Wi-Fi performance studies. [1] employed signal mapping and performance evaluations, revealing that library access points were frequently overloaded, with some supporting more than 40 simultaneous connections. At certain entry points, signal strength regularly fell below –70 dBm, rendering the network unreliable. To support such analysis, tools like Ekahau, NetSpot, and Vistumbler have been widely used, with [12] noting Vistumbler's GPS-based mapping features as particularly useful for large-scale campus coverage assessments.

Performance testing tools were also central to Wi-Fi evaluation. [13] compared iPerf, Netperf, and other utilities, reporting that iPerf achieved TCP throughput of up to 93.1 Mbps on a 100 Mbps link. This demonstrated its effectiveness in measuring key parameters such as bandwidth, latency, and packet loss, making it especially valuable for campus network evaluations.

At the University of Ghana, [**?**] implemented a supplementary public Wi-Fi solution using Mikrotik routers. Results showed a dramatic improvement: download speeds increased from 3–5 Mbps to over 20 Mbps, and signal strength improved by more than 15 dB in several test locations.

Together, these studies confirm that while Wi-Fi enables critical academic engagement, recurring challenges of coverage, bandwidth, interference, and user satisfaction remain notable. This positions the present study at the University of Zululand's KwaDlangezwa campus as a critical continuation of these efforts aimed at evaluating real-world Wi-Fi performance and recommending practical, context-specific improvements to strengthen student connectivity.

III. Research Method

A. Research Design

This study used a case study design with a mixed-methods approach to assess the performance of the UNIZULU-STUDENTS Wi-Fi network at the KwaDlangezwa campus. The case study structure allowed for a focused, in-depth analysis of this particular environment, centering on identifying coverage gaps, performance limitations, and student experiences (see Figure 1). By combining quantitative measurements with qualitative feedback, the study ensured that both technical and user-centered perspectives were captured. This dual emphasis aligned directly with the research goal of generating useful knowledge that can be applied in future initiatives to strengthen and improve the reliability of the campus Wi-Fi network operations.

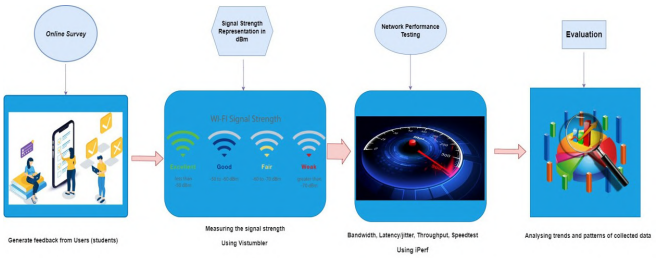


Fig. 1: Proposed architecture

B. Data Collection

Data was collected from three main sources: an online student survey, Wi-Fi coverage mapping with Vistumbler, and performance testing with iPerf3. The online survey was created in Microsoft Forms and distributed via WhatsApp groups, departmental lecturers, and faculty secretaries. Ultimately, 124 students responded, providing valuable insights into usage habits, levels of satisfaction, and frequently reported connectivity problems. The survey combined structured rating-scale questions with open-ended items, capturing both measurable data and richer student feedback, enabling a comprehensive understanding of user perceptions.

Vistumbler, an open-source Wi-Fi analyzer, was then used to gather technical measurements across multiple locations. It recorded signal strength (RSSI), interference, channel usage, and SSID availability. Signal strength was computed using the logarithmic formula shown in Equation (1), where P_{rx} is the received signal power in milliwatts (mW) [14]. In practice, –30 dBm indicates excellent signal strength, while values weaker than –70 dBm suggest unreliable coverage. During site surveys, hotspots in areas such as the library and residences frequently measured between –70 dBm and –80 dBm, confirming student reports of poor connectivity. Although Vistumbler supports GPS-based mapping, configuration issues with COM port detection meant that manual location referencing had to be used instead. Vistumbler was chosen over alternatives such as Ekahau or NetSpot because it is free, lightweight and integrates GPS functionality, making it particularly suitable for resource-constrained environments like public universities. Signal strength in dBm was computed:

$$\mathrm{RSSI(dBm)} = 10.\log_{10}\left(\frac{P_{rx}}{1\ \mathrm{mW}}\right) \tag{1}$$

Complementing this, iPerf3 was used to measure throughput, latency, jitter, and packet loss. Throughput was calculated using the formula shown in Equation (2), which captures the effective data rate between client and server [15], [16]. For example, Library tests peaked at ∼7.3 Mbps but dropped to averages as low as 439 Kbps, showing significant instability. Throughput:

$$\text{Throughput (Mbps)} = \frac{\text{Total Data Transferred (Mb)}}{\text{Time (s)}} \quad (2)$$

Latency, defined as the delay in transmitting packets, was expressed using the formula shown in Equation (3), [17]: Latency:

$$\text{Latency (ms)} = t_{\text{arrival}} - t_{\text{departure}} \quad (3)$$

While jitter, the variation in packet delay, was computed using the formula shown in Equation (4), [18], where d_i represents the one-way delay of packet i. Jitter values above 30 ms were considered disruptive for real-time activities such as video conferencing. Jitter:

$$\text{Jitter (ms)} = \frac{1}{N-1} \sum_{i=1}^{N-1} |d_{i+1} - d_i| \quad (4)$$

Finally, Packet loss which was the percentage of data packets that fail to reach their destination, was calculated using the formula shown in Equation (5) [19], UDP tests in the library recorded packet loss rates as high as 68%, pointing to heavy congestion and unreliable performance. iPerf3 was selected instead of alternatives such as Netperf or D-ITG because it is widely adopted in both academic and industry settings, it supports both TCP and UDP testing and offers precise control over test parameters, making it well-suited for benchmarking wireless networks. Packet Loss:

$$\text{Packet Loss (\%)} = \frac{\text{Packets Sent} - \text{Packets Received}}{\text{Packets Sent}} \times 100 \quad (5)$$

Ethical considerations were integrated into the data collection process. Survey participation was voluntary and based on informed consent. Responses were anonymized, and no personal identifiers were collected. All survey data were stored securely on Microsoft OneDrive, and technical measurements from Vistumbler and iPerf3 were used strictly for research purposes. The study also complied with institutional requirements by restricting testing to the official student Wi-Fi network.

C. Implementation

The study was carried out in three stages: survey deployment, Wi-Fi scanning, and performance testing.

In the first stage, distributing the survey to a large and diverse student population proved challenging. Initial response rates were low due to restricted access to mailing lists managed by the Student Representative Council (SRC). This hurdle was overcome by working closely with departmental lecturers and faculty secretaries and issuing follow-up reminders, which eventually increased responses to 124.

The second stage involved using Vistumbler to validate survey-reported weak signal areas. While effective in identifying coverage gaps, the GPS integration feature failed on the host laptop due to COM port recognition errors and as a workaround, manual referencing was employed. Although this reduced the ability to export results into Google Earth for visualization, it did not compromise the accuracy of the RSSI data collected.

In the final stage, iPerf3 was used for throughput and latency testing between a client laptop and a server in the University's Computer Science Honors Laboratory. A smartphone was also configured as a client to test mobile scenarios. However, tests conducted at greater distances from the University's Computer Science Honors Laboratory often failed due to weak Wi-Fi signals or firewall restrictions, leading to error messages such as "unable to connect to server." Rehosting the iPerf server on a laptop without firewall limitations resolved the problem and stabilized the testing process.

Despite these operational setbacks, the methodology successfully combined survey data with quantitative performance measurements. By applying established formulas for RSSI, throughput, latency, jitter, and packet loss, the study produced not only descriptive findings but also quantifiable evidence of Wi-Fi performance. This approach ensured both scientific rigor and practical relevance for addressing the connectivity challenges at the University of Zululand.

IV. Results and Discussion

The survey (124 responses) highlighted persistent dissatisfaction with the UNIZULU-STUDENTS Wi-Fi. Figure 2 and 3 shows that most students rated speed and reliability between Dissatisfied and Satisfied, with reliability (39%) scoring slightly higher than speed (32%). While Figure 4 shows that connectivity disruptions were common: 87% of students had at least occasional outages, while 12% experienced them consistently.

Coverage performance was location dependent. Figure 5 and 6 shows how the labs and departmental buildings performed somewhat better (34% acceptable, 24% good), but residences and sports grounds were rated poorly, with nearly 70% opting for poor or very poor. Students relied heavily on Wi-Fi for academic pursuits such as use of Moodle and e-resources (51% combined), alongside communication and streaming. Over 80% of respondents viewed high-speed Wi-Fi as important or very important for academic and extracurricular activities. These findings mirror those of [11], who also found responsiveness and reliability to be the strongest predictors of student satisfaction in Malaysia.

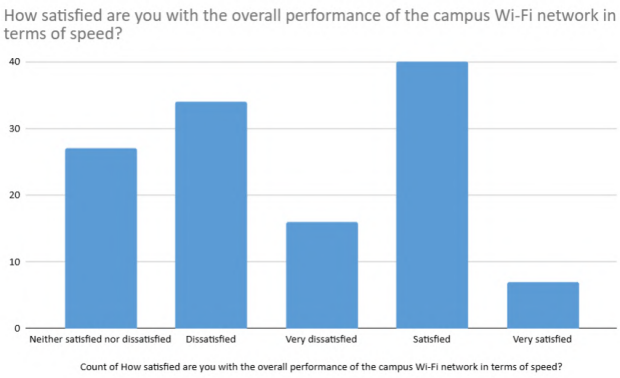


Fig. 2: Network Performance (speed) rating

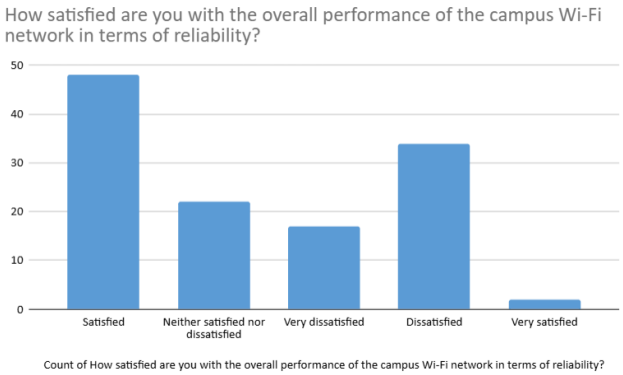


Fig. 3: Network Performance (Reliability) rating

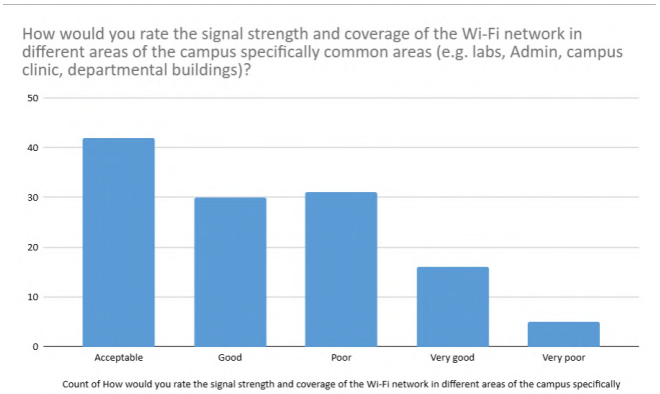


Fig. 5: Rating Signal Strength in different locations (Labs, Admin, campus clinic, department buildings)

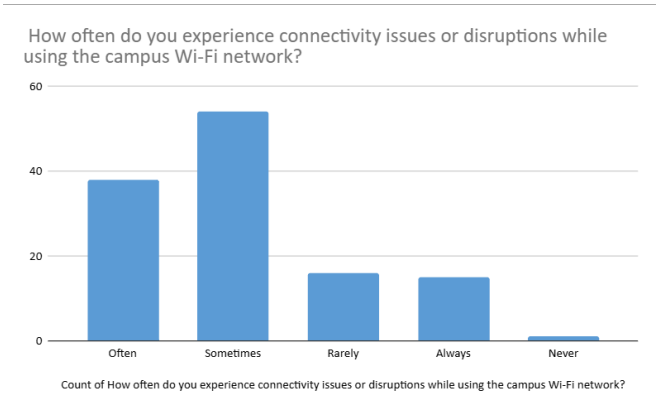


Fig. 4: Frequency of connectivity issues

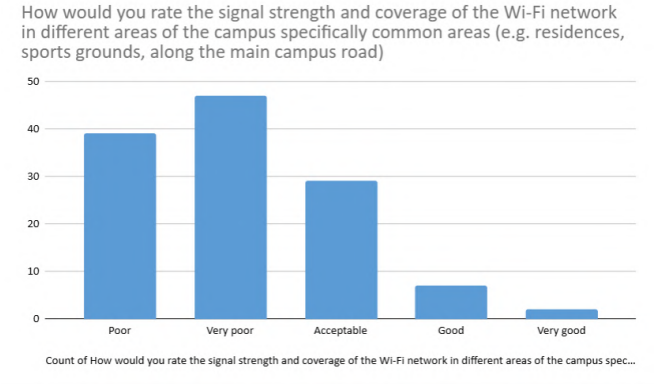


Fig. 6: Rating Signal Strength in different locations (Residences, sports ground, along the main campus road)

Vistumbler mapping substantiated these perceptions. Figure 7 shows the second floor of the library, where RSSI values fluctuated between –100 dBm to –79 dBm, well below the threshold for a stable service. Similar weaknesses were observed in the East residences, shown in Figure8, where RSSI values ranged from –77 dBm to –64 dBm, indicating unstable connections. In the lecture theatres (LTs), shown in Figure 9, 70 of 118 identified access points were active, but many produced weak signals (–75 dBm to –67 dBm). In some areas, competing networks had a higher output than the UNIZULU-STUDENTS SSID, often a sign of inadequate AP distribution and interference. These findings are consistent with those of [10], who noted comparable congestion and coverage gaps in universities in Ghana. Performance testing with iPerf3 also provided further evidence of instability. Under controlled conditions in the University's Computer Science Honors Laboratory, throughput remained stable at ∼300 Mbps between a Wi-Fi client and a wired Ethernet server, confirming that the backbone infrastructure was robust, which is shown in Figure 10. However, performance in real-world student facilities was far weaker. At the library, the TCP throughput averaged only 439 Kbps with repeated retransmissions (see Figure 11), while UDP tests yielded a jitter of 90.9 ms and a packet loss of 68% (see Figure 12). Such conditions were inadequate for real-time applications such as video conferencing, online lectures, or VoIP. Comparable observations were made by [13], who stressed the need to balance infrastructure capacity with configuration to ensure consistent throughput. Overall, the results demonstrated a clear gap between the capacity of the core infrastructure and student's actual connectivity experience. The University's Computer Science Honors Laboratory tests confirmed that the backbone network could deliver high throughput; however, poor placement of the access points, congestion, and interferences resulted in degraded performance in high-density areas such as the library, residential areas, and lecture theatres. Both survey data and technical assessments pinpoint the same key areas, highlighting the urgent need for targeted reconfiguration and coverage improvements to align performance with student expectations.

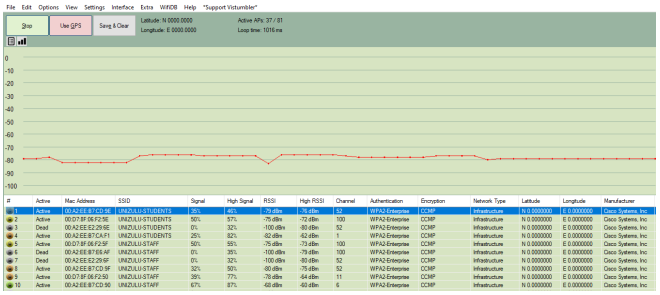


Fig. 7: Library Signal Coverage

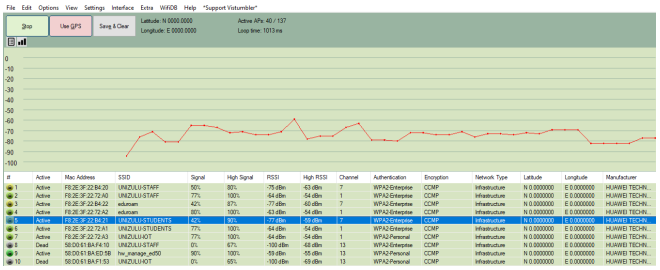


Fig. 8: East Residences signal coverage

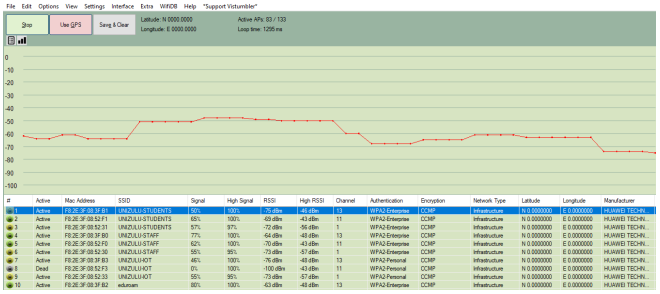


Fig. 9: LT'S Lecture theatres Signal Coverage

The findings of this study also indicate several targeted recommendations for improving the campus Wi-Fi network. Areas where RSSI values consistently fall between -70 dBm and -100 dBm should be prioritized for access point repositioning, while a coordinated channel allocation plan is needed to minimize overlap and interference between neighboring access points. Expanding the use of the 5 GHz band would help reduce congestion, and managing client load per access point could enhance performance in high-density locations such as the library and lecture theatres. Implementing bandwidth management strategies during peak usage periods may further stabilize throughput and reduce packet loss. Finally, adopting periodic monitoring practices such as scheduled Vistumbler scans and routine iPerf3 tests would enable the ICT staff to track performance trends and make informed adjustments over time.

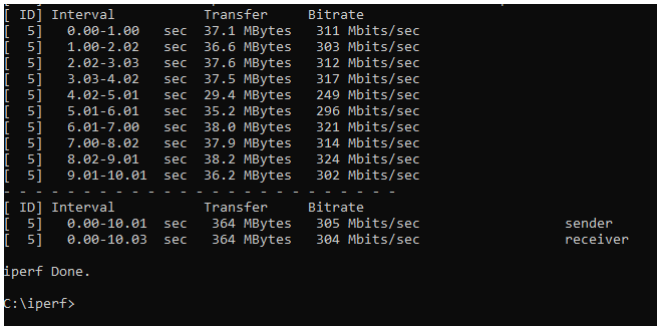


Fig. 10: iPerf client (Computer Science Honors Laboratory test)

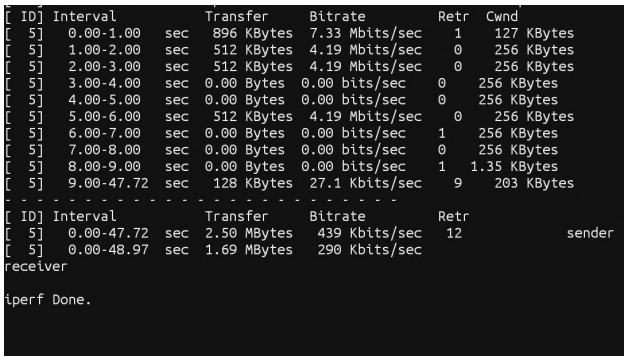


Fig. 11: TCP mode throughput and bandwidth tests at the library

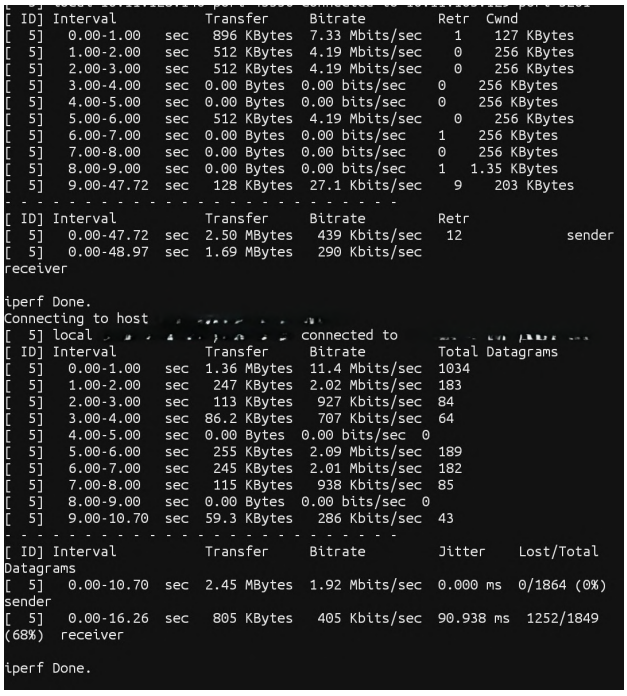


Fig. 12: UDP mode Latency, Jitter, and Packet loss tests at the library

V. Conclusion

In this work, we assessed the performance of the UNIZULU-STUDENTS Wi-Fi network at the University of Zululand's KwaDlangezwa campus, combining student perceptions with technical measurements. The survey highlighted notable dissatisfaction, with students reporting slow speeds, frequent disruptions, and poor coverage in critical areas such as the library, lecture theatres, residences, and sports grounds. Over 80% of respondents viewed high-speed Wi-Fi as important or very important for academic and extracurricular activities, emphasizing the need for the network's central role in supporting teaching, learning, and communication.

Vistumbler mapping confirmed these perceptions, showing unstable and weak signals, often ranging between –79 dBm and –100 dBm in critical areas. The findings suggested that access points were poorly positioned and subject to interference from overlapping channels, resulting in inconsistent connectivity and reduced reliability.

Performance testing with iPerf3 revealed further limitations. While controlled tests in the University's Computer Science Honors Laboratory achieved stable throughput at $\sim$300 Mbps, field tests at the library exposed poor bandwidth utilization, with average TCP throughput at just 439 Kbps, high retransmissions, and severe UDP performance issues, including jitter of 90.9 ms and packet loss of 68%. These metrics make the network unsuitable for real-time applications such as video conferencing, VoIP, or online lectures.

Overall, the study concludes that the campus Wi-Fi suffers from weak access point distribution, congestion, and interference, undermining its ability to meet student needs. While the backbone infrastructure can support high transfer rates, inefficiencies in deployment and configuration result in poor student experiences. Immediate improvements in access point placement, bandwidth management, and interference reduction are essential to support academic activities. Reliable Wi-Fi is no longer optional; it is central to student success and campus productivity. Addressing these gaps will allow the University of Zululand to provide a more robust digital environment that empowers students and meets the demands of modern Higher Education.

Future Work—While this study identified key gaps in coverage, stability, and access point placement, further work is needed to evaluate optimization strategies in practice. Future studies should incorporate automated data logging mechanisms to capture network performance trends over extended periods, allowing for a more progressive and detailed understanding of temporal variations in Wi-Fi quality. Additionally, future surveys should include demographic segmentation such as faculty, device type, and location to better contextualize user experience and identify usage patterns across different student groups. Further work should involve access point (AP) reconfiguration and optimization testing based on the findings of this study, including evaluating alternative AP placements, channel allocation techniques, and bandwidth management policies. Finally, incorporating predictive models and simulation

tools could provide deeper insights into how the network might behave under higher-demand scenarios, facilitating evidence-based improvements for campus-wide wireless infrastructure.

ACKNOWLEDGMENT

We acknowledge the support of the Telkom Centre of Excellence programme and the Department of Computer Science at the University of Zululand. This work was supported in part by the EWSETA bursary programme.

REFERENCES

[1] F. B. K. Easha, R. Abbas, and M. Daley, "Campus wi-fi coverage mapping and analysis," *arXiv preprint arXiv:2004.01561*, 2020.

[2] K. K.-K. Fong and S. K. S. Wong, "Wi-fi adoption and security in hong kong," *International Business Research*, vol. 10, no. 8, pp. 129–48, 2017.

[3] I. Shayea, M. Ergen, M. H. Azmi, S. A. Çolak, R. Nordin, and Y. I. Daradkeh, "Key challenges, drivers and solutions for mobility management in 5g networks: A survey," *IEEE access*, vol. 8, pp. 172 534–172 552, 2020.

[4] K. N. Ohei and R. Brink, "The effectiveness of wi-fi-network technology on campuses and residences for an improved learning experience and engagement," *Mousaion*, vol. 39, no. 1, 2021.

[5] M. Koziol, "Wi-fi's big bet on reliability: The next generation of the wireless tech retackles a common complaint," *IEEE Spectrum*, vol. 61, no. 01, pp. 48–49, 2024.

[6] Q. Qu, B. Li, M. Yang, Z. Yan, A. Yang, D.-J. Deng, and K.-C. Chen, "Survey and performance evaluation of the upcoming next generation wlans standard-ieee 802.11 ax," *Mobile Networks and Applications*, vol. 24, pp. 1461–1474, 2019.

[7] V. Mbonye, "Ukzn westville students' use of on-campus wi-fi and their perceptions of quality of service," Master of Commerce (Information Systems and Technology) Thesis, University of KwaZulu-Natal, College of Law and Management Studies, Durban, South Africa, 2019, master of Commerce (Information Systems and Technology). Supervisor: C. S. Price.

[8] F. Gururaj, H. Kumar, and M. Lokesha, "Wireless network use and awareness of surana college students: A study," *An International Peer Reviewed Bilingual E-Journal of Library and Information Science*, vol. 3, no. 2, pp. 70–77, 2016.

[9] Y. Han, "The study of students' perceptions of on-campus wireless local area networks (wlans) usage," Master of Computing Thesis, UNITEC New Zealand, Auckland, New Zealand, Aug. 2008, supervised by Dr. Logan Muller and Assoc. Dr. Sam Kolah.

[10] O. Nyarko-Boateng, I. K. Nti, A. F. Adekoya, B. A. Weyori, and S. Appiah, "Conceptual framework for efficient 802.11 a/b/g/n wireless networks infrastructure to improve campus internet connectivity," *International Journal of Engineering Technologies and Management Research*, vol. 7, no. 11, pp. 1–8, 2020. [Online]. Available: https://doi.org/10.29121/ijetmr.v7.i11.2020.802

[11] N. A. N. Ismail, N. Nazmi, N. H. B. A. Razak, N. A. Azhar, R. Shaari, and M. S. Omar, "Influence of wi-fi performance on students' satisfaction," *International Journal of Modern Education*, vol. 3, no. 8, pp. 226–237, mar. 2021. [Online]. Available: https://www.ijmoe.com

[12] M. Jivthesh, M. Gaushik, P. Adarsh, G. H. Niranga, and N. S. Rao, "A comprehensive survey of wifi analyzer tools," in *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)*. Bangalore, India: IEEE, Oct. 2022, Conference Proceedings, pp. 1–8.

[13] O. Olimov, G. Artikova, and M. Xatamova, "Iperf to determine network speed and functionality," *Web of Technology: Multidimensional Research Journal*, vol. 2, no. 3, pp. 94–101, 2024.

[14] A. Buchman and C. Lung, "On the relationship between received signal strength and received signal strength index of ieee 802.11 compatible radio transceivers," *Carpathian Journal of Electronic and Computer Engineering*, vol. 6, no. 2, pp. 15–20, 2013. [Online]. Available: http://cjece.ubm.ro

[15] R. Eliviani, "Throughput prediction for multimedia iot in wireless network," in *2022 International Symposium on Electronics and Smart Devices (ISESD)*. IEEE, 2022, Conference Proceedings, pp. 1–6, print on Demand ISBN: 979-8-3503-9661-4.

[16] Meter.com, "How to calculate throughput in your network-the right way," www.meter.com, 24 February 2025 2024, accessed August 2025. [Online]. Available: https://www.meter.com/resources/

[17] A. Model, "Network latency," www.analytics-model.com, 2020, accessed August 2025. [Online]. Available: https://www.analytics-model.com/

[18] I. Daly, M. Novotny, Z. Hajaiej, and A. Garsallah, "Accuracy of jitter and shimmer measurements for speaker in the database timit and ntimit," in *2016 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*. IEEE, July 2016, Conference Proceedings, pp. 1–5.

[19] J. Birla, B. Sah, and B. BRCM, "Performance metrics in ad-hoc network," *BRCM, Bahal*, 2012.

Multi-Objective Optimization for Trust-Based Relay Selection in 6G-IoT Sidelink Communication

Stephen O. Ekwe, Joyce B. Mwangama
Department of Electrical Engineering, *University of Cape Town*
Cape Town, South Africa
ekwste001@myuct.ac.za, joyce.mwangama@uct.ac.za

***Abstract*—Device heterogeneity in IoT ecosystems introduces complex challenges for data aggregation and forwarding, particularly in selecting reliable relay nodes and managing scarce radio resources. Many existing approaches optimize a single performance metric or rely on static thresholds, overlooking critical trade-offs such as energy efficiency, latency, and device lifetime. Furthermore, they often fail to ensure reliable and trustworthy connections, especially in sidelink communication. To address these gaps, we propose a trust-based relay selection framework that integrates fuzzy logic to model device reliability from historical interactions. In doing so, the framework leverages the social structures that naturally emerge from the daily interactions of network devices, enabling more reliable and adaptive cooperation. The problem is formulated as a multi-objective mixed-integer nonlinear program (MO-MINLP), which is inherently non-convex and NP-hard. To solve it, we employ the Deep Sleep Optimizer (DSO), a recent metaheuristic that balances exploration and exploitation through sleep–wake cycles, and benchmark against SQP, PSO, and GWO. Simulation results demonstrate that the proposed framework achieves better trade-offs, enabling energy-efficient and socially reliable sidelink communication among heterogeneous IoT devices in dense 6G networks.**

***Index Terms*—MO-MINLP, social-aware, relay selection, DSO**

I. Introduction

The evolution of fifth-generation (5G NR) and the ongoing development of sixth-generation (6G) networks, together with the proliferation of Internet of Things (IoT) devices, are enabling new opportunities for smart cities. These technologies promise ultra-fast connectivity, automation, and intelligent services, supporting applications ranging from immersive AR/VR to real-time monitoring for healthcare, transportation, and energy management [1]. In addition to consumer services, 3GPP has emphasized the importance of device-to-device (D2D) and sidelink communication for *public safety, emergency response, and extended coverage* scenarios, where reliable connectivity between user equipments (UEs) is crucial when infrastructure is overloaded or unavailable [2], [3]. However, the unprecedented scale of IoT is expected to exceed 97 billion connected devices by 2030, combined with stringent 6G targets of 1 Tbps peak rates, 0.1 ms latency, and ultra-reliability (10^{-9}) [4], introduces significant challenges in resource management and sustainability.

This work was supported by TELKOM SA via the Telkom Centre of Excellence (CoE) in Broadband Networks and Application at the University of Cape Town as well as the SENTECH Research and Innovation Hub

Relay selection has emerged as a central bottleneck in dense 6G-IoT environments. FCC's Spectrum Horizons First Report and Order established experimental licensing for frequencies between 95 GHz and 3 THz to foster innovation in high-frequency communications, while ITU-R has identified bands above 100 GHz as candidates for IMT-2030 [5], [6]. These sub-THz and very-high-frequency bands enable ultra-dense deployments due to their small cell radius and large available bandwidth. However, they also incur high propagation loss and require highly directional antennas and beamforming, which increase coordination overhead. As a result, achieving an effective balance among throughput, latency, energy efficiency, and device lifetime necessitates relay strategies that can manage frequent link evaluation, beam alignment, and signaling overhead. Recent studies confirm that exhaustive or naive selection in such conditions leads to excessive overhead, motivating more adaptive approaches [7]–[9].

Early contributions on relay selection primarily optimized single metrics such as channel quality or residual energy [10]. Subsequent work introduced social-awareness, where relations such as reputation, distance, or interaction frequency were used to enhance cooperation among devices [11]. While these models improved fairness, they often relied on fixed thresholds or static trust values and did not jointly address multiple performance objectives. Other efforts explored fuzzy-logic-based techniques, for instance in routing for wireless sensor networks [12] and secure relay selection for IoT systems [13].

A complementary research line considered the impact of coordination overhead in high-frequency bands. Lou *et al.* [7] examined hybrid RF/THz relay selection as a means to avoid full channel state information (CSI) acquisition, while Kim *et al.* [8] used deep reinforcement learning to jointly manage relay selection and beam training in vehicular mmWave networks. Iwabuchi *et al.* [9] demonstrated through experimental trials and simulations that distributed MIMO and RIS-assisted relay beamforming can substantially improve throughput and coverage.

Given the combinatorial and nonconvex nature of relay selection and resource allocation, recent studies have increasingly employed meta-heuristic and multi-objective optimization frameworks to balance conflicting performance goals. Ayedi *et al.* [14] proposed an improved Salp Swarm Algorithm that jointly optimizes energy efficiency and network lifetime in multi-relay underground sensor networks. Babu *et al.* [15]

developed a hybrid framework that combines Manta Ray Foraging Optimization with deep learning for joint relay selection and resource allocation in D2D communications. Raj *et al.* [16] introduced an elephant-herd-inspired approach integrated with neuro-fuzzy trust modeling for clustering and relay selection in MANETs. Beyond terrestrial systems, Mohammadi *et al.* [17] investigated joint relay placement and selection in underwater acoustic networks, explicitly targeting network lifetime maximization, while Zhou *et al.* [18] examined relay selection for over-the-air computation with lifetime-aware design.

A. Contributions

This paper proposes the *Social-Aware Relay Selection (SARS)* algorithm, a trust-based and fuzzy-logic-driven framework for relay selection in 6G-IoT sidelink communication. The problem is cast as a multi-objective nonconvex program and solved using the Deep Sleep Optimizer (DSO), with extensive comparative evaluation. The main contributions are summarized as follows:

- Development of the SARS algorithm, formulated as a multi-objective mixed-integer nonlinear program (MO-MINLP) for dense 6G-IoT sidelink networks, jointly optimizing energy efficiency, latency, spectral efficiency, and network lifetime.
- Design of a fuzzy-logic trust inference model that exploits historical data, capturing distance, frequency of interaction, and satisfaction coefficients of past associations, to enable socially reliable community formation.
- Introduction of a hybrid trust architecture in which user equipments (UEs) maintain local trust values while periodically reporting to the base station for community-level optimization.
- Application of the Deep Sleep Optimizer (DSO) to solve the nonconvex MO-MINLP, with SARS benchmarked against SQP, PSO, and GWO. Results show better Pareto trade-offs across all considered metrics, highlighting the effectiveness of SARS for socially reliable and energy-efficient communication.

The remainder of the paper is organized as follows. Section II describes the physical layer, social network, and community-formation models for 6G-IoT sidelink communication. Section III formulates the multi-objective relay-selection problem and presents the proposed SARS framework together with the DSO-based solution method. Section IV reports numerical results and comparative analysis against baseline schemes. Section V concludes the paper and outlines directions for future work.

II. System Model

We consider a 6G-enabled smart city network comprising dense small-cell base stations (SCBs) and heterogeneous IoT devices capable of device-to-device (D2D) communication. Relay-assisted cooperation is supported through the 3GPP Proximity Services (ProSe, Rel-12) [19], where devices can operate as intermediate nodes to extend coverage and enhance reliability. Extending this capability, our model incorporates the 5G New Radio (NR) sidelink introduced in Rel-17 [20], which enables direct UE-to-UE data transfer and relay functionality.

A. Device heterogeneity

The IoT ecosystem in this model spans low-power sensors, wearables, and environmental monitors, as well as high-performance devices such as AR/VR headsets, autonomous vehicles, and holographic communication systems. Each device $u \in \mathcal{U}$ is modeled by a capability vector defined as

$$\mathbf{A}_u = [\delta_u, \beta_u, \pi_u, \rho_u, \psi_u], \tag{1}$$

where δ_u denotes the device type, β_u battery capacity, π_u its QoS requirement, ρ_u processing power, and ψ_u its service priority.

B. Mobility.

Devices follow one of four mobility models: static (e.g., traffic lights), random walk (e.g., pedestrians), Lévy flight (e.g., drones), and high-speed vehicular motion. These mobility patterns reflect practical deployments in urban IoT scenarios and correspond to the mobility classes considered in NR sidelink evaluations for vehicular and public safety communication [21]. Incorporating mobility into our relay model ensures robustness under dynamic topologies.

C. Sidelink Access and Discovery.

In accordance with 3GPP specifications, devices may establish sidelink communication in two modes: (i) *network-scheduled*, where the base station (gNB/SCB) allocates resources and assists in pairing UEs for D2D communication; and (ii) *autonomous*, where devices discover and select resources directly without base station involvement [19], [22]. In this work, we consider *in-band direct association*, where devices form cooperative groups over the same carrier resources as the cellular link. Although BS-assisted pairing offers centralized coordination and reduced collision risk, our focus is on *direct community formation* to emphasize the social-aware aspect of cooperation. This reflects ProSe-style autonomous discovery, where UEs broadcast and listen on sidelink discovery channels to identify potential peers.

D. Physical Network Model

We consider downlink (DL) and uplink (UL) transmissions in a dense small-cell network with base stations (SCBs) and heterogeneous IoT devices capable of relay-assisted D2D communication. Let $\mathcal{M}$ denote the set of SCBs and $\mathcal{U}$ the set of IoT devices. Devices are grouped into social-aware cooperative communities $\mathcal{K}$, where a subset $\mathcal{U}_k^{\mathrm{R}} \subset \mathcal{U}_k$ acts as relays, while the remaining devices $\mathcal{U}_k^{\mathrm{D}} = \mathcal{U}_k \setminus \mathcal{U}_k^{\mathrm{R}}$ rely on them for connectivity. For each SCB m, we denote by $\mathcal{U}_m$ the set of relay devices it serves, which is the union of relay sets of the communities attached to m, i.e., $\mathcal{U}_m = \bigcup_{k \in \mathcal{K}_m} \mathcal{U}_k^{\mathrm{R}}$, where $\mathcal{K}_m \subseteq \mathcal{K}$ is the set of communities associated with SCB m.

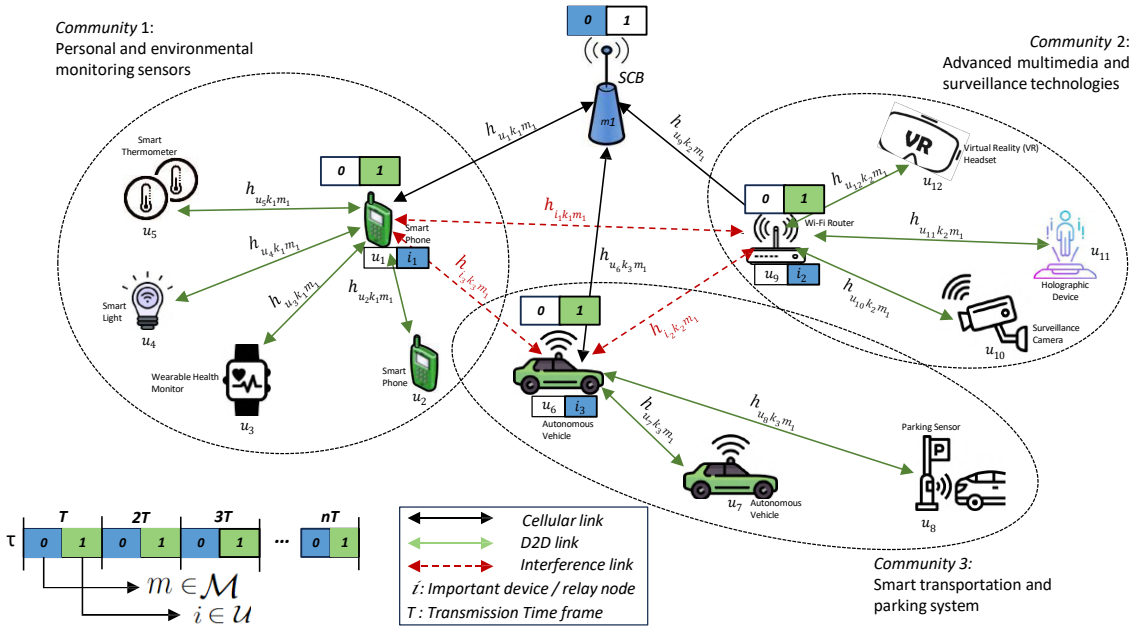


Fig. 1: System model of relay-assisted sidelink communication in 6G-IoT small cell networks.

The network operates at $f_c = 150\,\text{GHz}$ with bandwidth $B = 10\,\text{GHz}$. Large-scale propagation follows the Close-In (CI) path-loss model:

$$PL_{x,y} = FSPL(f_c, d_0) + 10n\log_{10}\left(\frac{d_{x,y}}{d_0}\right) + \chi_\sigma, \qquad (2)$$

where $d_{x,y}$ is the 3D Tx–Rx distance, n the path-loss exponent, and $\chi_\sigma \sim \mathcal{N}(0, \sigma^2)$ the log-normal shadowing term. The corresponding channel gain is $h_{x,y} = 10^{-PL_{x,y}/10}$, and the received power is

$$P_r(x \to y) = P_t + G_t + G_r - PL_{x,y} - L_{\text{misc}}. \qquad (3)$$

Here, P_t denotes the transmit power (dBm), G_t and G_r are the transmit and receive antenna gains (dBi), $PL_{x,y}$ is the path-loss (dB) between nodes x and y, and L_{misc} accounts for additional fixed system losses (e.g., hardware or polarization mismatch).

We employ scalable OFDM numerology with subcarrier spacing $\Delta f_\mu = \Delta f \cdot 2^\mu$, $\Delta f = 15\,\text{kHz}$. For $\mu = 6$, the subcarrier spacing is $960\,\text{kHz}$, yielding an OFDM symbol duration $T = 1.04\,\mu\text{s}$ [23].

Relay-assisted communication follows a periodic frame structure. Each frame has duration T_f and is divided into four sub-phases: (i) DL phase 0 (SCB-to-relay), (ii) DL phase 1 (relay-to-device), (iii) UL phase 0 (device-to-relay), and (iv) UL phase 1 (relay-to-SCB). The normalized phase durations satisfy

$$\tau_0^{\text{DL}} + \tau_1^{\text{DL}} + \tau_0^{\text{UL}} + \tau_1^{\text{UL}} = 1. \qquad (4)$$

Let $\mathcal{U}_m \subseteq \mathcal{U}$ denote the set of relays scheduled by SCB m. The total bandwidth B is partitioned across relays as $B_{m,i}$, and the DL transmit power of SCB m toward relay u_i is denoted $P_{m,i}^{\text{DL}}$. The UL transmit power of device u_j is $P_{u_j}^{\text{UL}}$. The bandwidth and power budgets per SCB are constrained by

$$\sum_{i \in \mathcal{U}_m} B_{m,i} \le B, \qquad \forall m \in M,$$
$$\sum_{i \in \mathcal{U}_m} P_{m,i}^{\text{DL}} \le P_m^{\max}, \qquad \forall m \in M,$$

where $P_m^{\max}$ is the maximum transmit power of SCB m. For tractability and to focus on relay selection rather than fine-grained resource scheduling, we assume equal allocation per relay: $B_{m,i} = B/|\mathcal{U}_m|$ and $P_{m,i}^{\text{DL}} = P_m^{\max}/|\mathcal{U}_m|$, which is consistent with OFDMA-based sidelink resource partitioning in NR systems [22]–[24].

We explicitly model both downlink (DL) and uplink (UL) two-hop transmissions, where SCB m communicates with device u_j via relay u_i in each direction. The channel, interference, and resource-allocation models introduced below apply symmetrically to DL and UL, and the end-to-end rate $R_{u,k,m}$ is defined by the weaker of the two hops.

The achievable hop rates in DL phase 0 and UL phase 1 are then given by

$$R_{u_i,k,m}^{\text{DL}} = \frac{\tau_0^{\text{DL}}}{T} B_{m,i} \log_2\left(1 + \frac{h_{u_i,k,m} P_{m,i}^{\text{DL}}}{N_0 + I_{u_i}^{\text{DL}}}\right), \qquad (5)$$

$$R_{u_j,k,m}^{\text{UL}} = \frac{\tau_1^{\text{UL}}}{T} B_{m,i} \log_2\left(1 + \frac{h_{u_j,k,m} P_{u_j}^{\text{UL}}}{N_0 + I_m^{\text{UL}}}\right), \qquad (6)$$

where T is the OFDM symbol duration introduced above, N_0 is the noise power spectral density, and $I_{u_i}^{\text{DL}}$ and I_m^{UL} are the

aggregate interference terms on the DL and UL, respectively. Using the same interference definitions as in (6)–(7), the end-to-end rate of device u in community k associated with SCB m is limited by the weaker hop:

$$R_{u,k,m} = \min\Big(R^{\mathrm{DL}}_{u_i,k,m}, R^{\mathrm{UL}}_{u_j,k,m}\Big). \tag{7}$$

Here, DL phases 0 and 1 correspond to SCB→relay and relay→device transmissions, respectively, while UL phases 0 and 1 capture device→relay and relay→SCB transmissions. This explicit frame structure will be used in Section III when defining latency, lifetime, and energy-efficiency metrics.

E. Social Network Model

We represent device interactions using a directed social graph $\mathcal{G} = (\mathcal{U}, \xi, \Gamma)$, where $\mathcal{U}$ is the set of IoT devices, $\xi \subseteq \mathcal{U}\times\mathcal{U}$ the directed links, and $\Gamma_{u,\tilde{u}} \in [0,1]$ the trust weight between devices u and $\tilde{u}$. To capture the uncertainty and multidimensional nature of social ties, we employ a Mamdani fuzzy inference system (FIS) [25]. The fuzzy inference system shown in figure (2) quantifies the social ties between devices by evaluating their historical data, using attributes such as distance similarity, frequency of interaction, and satisfaction coefficients. Rule activation strength is computed using the min–max composition method [26], while the centroid (CoG) defuzzification method [12] yields a crisp relationship score Ω. This score is mapped to discrete trust levels ranging from *Intruder* to *Spouse*, as defined in (8).

$$\Gamma_{u,\tilde{u}}(x) = \begin{cases} 0, & \Omega < 0.8, \quad \textit{Intruder}, \\ 0.2, & 0.8 \le \Omega < 1.2, \quad \textit{Acquaintance}, \\ 0.4, & 1.2 \le \Omega < 2.0, \quad \textit{Peer}, \\ 0.6, & 2.0 \le \Omega < 2.6, \quad \textit{Collaborator}, \\ 0.7, & 2.6 \le \Omega < 3.2, \quad \textit{Sibling}, \\ 0.85, & 3.2 \le \Omega < 3.8, \quad \textit{Parents}, \\ 1.0, & \Omega \ge 3.8, \quad \textit{Spouse}. \end{cases} \tag{8}$$

The resulting pairwise trust values form the trust matrix $\Gamma(x)$ in (9), where each entry $\Gamma_{u,\tilde{u}}(x)$ quantifies the reliability of device u with respect to $\tilde{u}$.

$$\Gamma(x) = \begin{bmatrix} \Gamma_{1,1}(x) & \Gamma_{1,2}(x) & \cdots & \Gamma_{1,\tilde{u}}(x) \\ \Gamma_{2,1}(x) & \Gamma_{2,2}(x) & \cdots & \Gamma_{2,\tilde{u}}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma_{u,1}(x) & \Gamma_{u,2}(x) & \cdots & \Gamma_{u,\tilde{u}}(x) \end{bmatrix}, \quad \forall u, \tilde{u} \in \mathcal{U}. \tag{9}$$

To evaluate trust at the device level, column-wise normalization produces the trust vector $\mathbf{T}$ in (10), which represents the relative trustworthiness of each device in the network. This structured quantification enables socially reliable relay selection by ensuring that stronger ties translate into higher trust, fostering cooperation and enhancing the robustness of 6G-IoT networks.

$$\mathbf{T} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{\mathcal{U}} \end{bmatrix}, \quad T_u = \frac{\sum_{\tilde{u}=1}^{|\mathcal{U}|} \Gamma_{u,\tilde{u}}(x)}{\sum_{u=1}^{|\mathcal{U}|}\sum_{\tilde{u}=1}^{|\mathcal{U}|} \Gamma_{u,\tilde{u}}(x)}, \quad u \in \mathcal{U}. \tag{10}$$

F. Community Formation

In the proposed framework, IoT devices form cooperative communities based on social trust and physical constraints. A symmetrized trust matrix $\bar{\Gamma} = \frac{1}{2}(\Gamma + \Gamma^{\top})$ defines strong ties when $\bar{\Gamma}_{u,\tilde{u}} \ge \tau$, where τ is the trust threshold. The resulting strong-tie graph is

$$W = \bar{\Gamma} \odot \mathbf{1}(\bar{\Gamma} \ge \tau), \quad \mathcal{U}_{\text{elig}} = \{u \in \mathcal{U} : \deg_W(u) > 0\}. \tag{11}$$

Communities are capacity-limited ($|\mathcal{U}_k| \le C_{\max}$) and constrained by maximum physical distance $d_{\max}$:

$$\mathcal{U}_k \subseteq \{u \in \mathcal{U}_{\text{elig}} : d(u, s_k) \le d_{\max}\}, \quad \mathcal{U}_k \cap \mathcal{U}_{k'} = \varnothing \ (k \ne k'). \tag{12}$$

Seed devices s_k are chosen using a composite score that balances closeness centrality, service priority, and battery level:

$$s(u) = \alpha_1 \hat{c}_u + \alpha_2 \psi_u + \alpha_3 \beta_u, \tag{13}$$

where $\hat{c}_u$, ψ_u, and β_u are normalized values of closeness centrality, service priority, and battery capacity, respectively, and the weight $\alpha_1 + \alpha_2 + \alpha_3 = 1$.

Members are assigned to the nearest trusted seed within $d_{\max}$ using trust-space shortest paths:

$$\mathcal{U}_k = \arg \min_{\substack{\mathcal{S} \subseteq \mathcal{B}(s_k, d_{\max}) \\ |\mathcal{S}| \le C_{\max}}} \sum_{u \in \mathcal{S}} D(s_k, u), \tag{14}$$

where $\mathcal{B}(s_k, d_{\max})$ is the candidate set and $D(s_k, u)$ is the shortest-path distance.

Finally, the relay head h_k is selected among community members with sufficient battery $\beta_u \ge \beta_{\min}$, maximizing the local composite score:

$$h_k = \arg \max_{u \in \mathcal{U}_k,\ b(u) \ge b_{\min}} \big(\alpha_1 \hat{c}_{u,k} + \alpha_2 \psi_{u,k} + \alpha_3 \beta_{u,k}\big). \tag{15}$$

This formulation ensures that only trusted and sustainable devices participate in cooperation, while head nodes balance connectivity, service priority, and energy availability. The resulting communities provide the foundation for the multi-objective optimization in Section III.

III. Problem Formulation

We formulate a multi-objective optimization framework for relay-assisted 6G-IoT networks, where the goal is to jointly optimize social welfare, network lifetime, energy efficiency, and latency. These metrics capture both the physical performance of the network and the social-awareness dimension introduced by trust modeling.

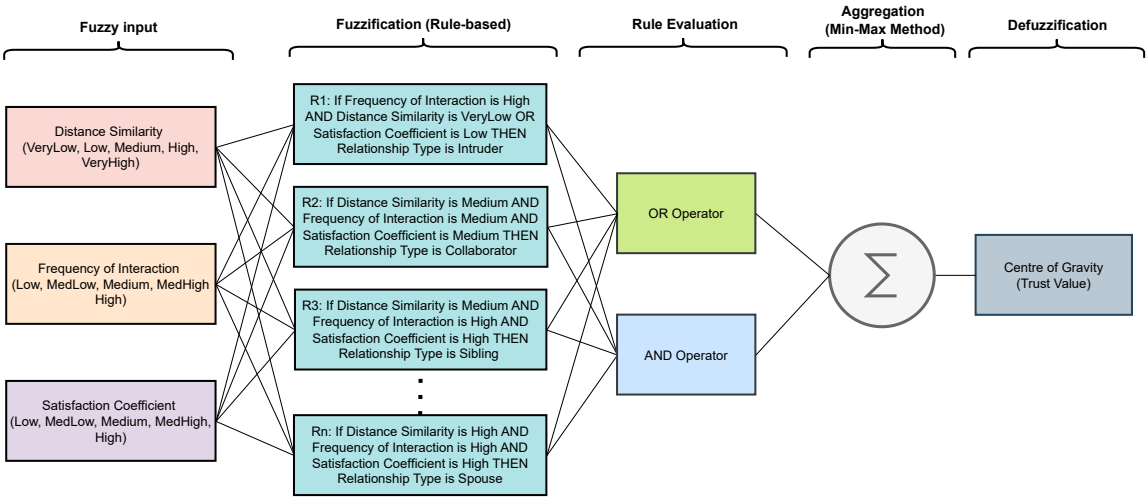


Fig. 2: Rule-based fuzzy logic trust model for social relationship classification.

Algorithm 1: Social-Aware Relay Selection (SARS) Framework

Input: Trust matrix Γ, distances $d(\cdot,\cdot)$, trust threshold τ, join radius $d_{\max}$, capacity $C_{\max}$, mininum battery $\beta_{\min}$, priority $\psi(\cdot)$, battery $\beta(\cdot)$

Output: Communities $\{\mathcal{U}_k\}$ and relay heads $\{h_k\}$

Symmetrize Γ and threshold by (11) ;
Compute shortest-path distances $D(\cdot,\cdot)$;
Compute closeness $c(u)$;
Normalize $\hat{c}(u), \psi_u, \beta_u$;
Compute seed score $s(u)$ by (13);;
Sort devices by $s(u)$;
$\mathcal{A} \leftarrow \mathcal{U}_{\text{elig}}$, $k \leftarrow 0$;
while $\mathcal{A} \neq \varnothing$ **do**
    Select next seed s_k with highest $s(u)$;
    Form $\mathcal{U}_k$ by (14) under constraints (12);
    $\mathcal{A} \leftarrow \mathcal{A} \setminus \mathcal{U}_k$; $k \leftarrow k+1$;
end
for $k = 1$ *to* K **do**
    Among $u \in \mathcal{U}_k$ with $\beta_u \geq \beta_{\min}$, select h_k by (15)u;
    If none feasible, select fallback device by same score;
end
return $\{\mathcal{U}_k\}$, $\{h_k\}$;

A. Social Welfare

In conventional networks, resource allocation is determined purely by channel or QoS constraints. However, in social-aware IoT networks, devices form relationships that affect cooperation and reliability. To capture this, we define social welfare as the combination of a device's trust score and its achievable end-to-end rate. Formally, the welfare of device u in community k is expressed as

$$\text{SW}_{u,k} = \mathbf{T}_u \cdot R_{u,k,m}, \tag{16}$$

where $\mathbf{T}_u$ is the normalized trust score from the social graph, and $R_{u,k,m}$ is the achievable rate in (7). The optimization objective maximizes aggregate welfare across all devices:

$$f_1 : \max \sum_{k\in\mathcal{K}} \sum_{u\in\mathcal{U}} \varphi_u \, \text{SW}_{u,k}, \tag{17}$$

where $\varphi_u \in \{0,1\}$ determines whether device u is allocated resources. This ensures that devices with higher social importance receive proportionally more resources, improving trust-driven cooperation.

B. Network Lifetime

Energy sustainability is a critical challenge in dense IoT deployments, as battery-limited devices can shorten overall system performance. We define the lifetime of a device as the ratio of its available battery to total power consumption. This accounts for downlink, uplink, and baseline operational demands:

$$L_u = \frac{\beta_u}{P_u^{Total}}, \quad P_u^{Total} = P_m^{DL} + P_u^{UL} + \rho_u^{base}. \tag{18}$$

Here, ρ_u^{base} denotes the processing power of the device, which ranges from a few milliwatts in low-power sensors to nearly 1 W in high-performance platforms such as autonomous vehicles. To ensure fairness, the effective lifetime $\bar{L}_u$ is bounded by both connectivity-based lifetime and critical device survival:

$$\bar{L}_u = \min(L_{\text{conn}}, L_{\text{crit}}). \tag{19}$$

Here, connectivity-based lifetime L_{conn} [27] represents the time until a specified fraction (τ) of devices depletes their energy:

$$L_{\text{conn}} = L_u(x), \quad \text{where} \quad x = \lceil |\mathcal{U}| \cdot (1-\tau) \rceil. \tag{20}$$

Similarly, critical device lifetime L_{crit} ensures the survival of relay nodes essential for network connectivity:

$$L_{\text{crit}} = \min\{L_u \mid u \in \mathcal{R}\}. \tag{21}$$

Thus, the lifetime objective adopts a max–min approach:

$$f_2 : \max \min_{u \in \mathcal{U}} (\varphi_u \bar{L}_u). \tag{22}$$

This guarantees that no single device becomes the bottleneck for network sustainability.

C. Energy Efficiency

While lifetime focuses on survival, energy efficiency (EE) captures how effectively devices use their energy in delivering data. In particular, EE is measured in bits-per-second per Watt, representing the trade-off between throughput and power consumption. For each device u in community k, we define:

$$\eta_{u,k,m} = \frac{R_{u,k,m}}{P_u^{Total}}. \tag{23}$$

This metric balances high data rates with low energy usage. The objective aggregates EE across the network with device-specific weights:

$$f_3 : \max \sum_{m,k,u} \varphi_u \zeta_{u,k,m} \eta_{u,k,m}, \tag{24}$$

where $\zeta_{u,k,m}$ prioritizes critical applications. In doing so, the framework ensures that energy-constrained devices remain sustainable without compromising system throughput.

D. Latency

Latency is a decisive factor in time-critical IoT applications, such as autonomous driving and AR/VR. We model latency as the ratio of packet size to the achievable rate:

$$\mathcal{L}_u = \frac{\mathcal{S}_u}{R_{u,k,m}}. \tag{25}$$

Devices with lower rates thus incur higher delays. To capture a fairness-oriented, network-wide objective we focus on the worst-case latency among all scheduled devices:

$$\bar{\mathcal{L}}_u = \max_{u \in \mathcal{U}} \mathcal{L}_u. \tag{26}$$

Our latency objective is then to minimise this quantity:

$$f_4 : \min \bar{\mathcal{L}}_u. \tag{27}$$

This min–max criterion ensures that no individual device experiences an excessively large delay, while real-time applications remain protected. To enforce feasibility, we also impose a hard constraint $\bar{\mathcal{L}}_u \leq \mathcal{L}_{max}$ for all devices.

To avoid unit inconsistencies and to combine all four objectives in a single scalar utility, we map each raw metric f_i (with $n = 1, \ldots, 4$ corresponding to (17), (22), (24), and (27)) into a dimensionless, maximization-oriented score $\tilde{f}_i \in [0, 1]$:

$$\tilde{f}_n = \frac{f_n - f_n^{\min}}{f_n^{\max} - f_n^{\min}}, \qquad n = 1, \ldots, 4. \tag{28}$$

Here, $f_n^{\min}$ and $f_n^{\max}$ denote lower and upper bounds for each metric, obtained from baseline schemes (e.g., CARS and EARS) or from known theoretical limits. In this way, all objectives are expressed as increasing utilities on a common scale, which is a standard requirement for weighted-sum scalarization in multi-objective optimization [28], [29].

E. Multi-Objective Formulation

The four normalized utilities $\tilde{f}_n$ are combined through a weighted-sum scalarization:

$$\max_{\boldsymbol{\varphi},\, \mathbf{P}^{\mathrm{DL}},\, \mathbf{P}^{\mathrm{UL}}} \sum_{n=1}^{4} \omega_n \tilde{f}_n. \tag{29}$$

subject to QoS, SINR, power, network lifetime, latency, and social constraints. The non-negative weights ω_n satisfy $\sum_{n=1}^{4} \omega_n = 1$ and encode application-specific priorities, e.g., ultra-reliable low-latency communication (URLLC) scenarios versus energy-focused massive machine-type communication (mMTC) deployments.

The complete optimization problem is

$$\sum_{m \in M} \sum_{k \in K} R_{u,k,m} \geq R_{\min}, \qquad \forall u \in \mathcal{U}, \tag{29a}$$

$$0 \leq P_u^{\mathrm{UL}} \leq P_u^{\max}, \qquad \forall u \in \mathcal{U}, \tag{29b}$$

$$0 \leq P_{m,i}^{\mathrm{DL}} \leq P_m^{\max}, \qquad \forall m \in M,\, i \in \mathcal{U}_m, \tag{29c}$$

$$\sum_{i \in U_m} P_{m,i}^{\mathrm{DL}} \leq P_m^{\max}, \qquad \forall m \in M, \tag{29d}$$

$$\gamma_u^{\mathrm{UL}} \geq \gamma_{\min}, \quad \gamma_u^{\mathrm{DL}} \geq \gamma_{\min}, \qquad \forall u \in \mathcal{U}, \tag{29e}$$

$$\bar{L}_u \geq \bar{L}_{\min}, \qquad \forall u \in \mathcal{U}, \tag{29f}$$

$$\mathcal{L}_u \leq \mathcal{L}_{\max}, \qquad \forall u \in \mathcal{U}, \tag{29g}$$

$$\sum_{n=1}^{4} \omega_n = 1, \quad \omega_n \geq 0, \qquad n = 1, \ldots, 4, \tag{29h}$$

$$\varphi_u \in \{0, 1\}, \qquad \forall u \in \mathcal{U}. \tag{29i}$$

Here, (29a) enforces a minimum average rate per device, (29b)–(29d) impose UL and DL power budgets (including the total transmit power per SCB), (29e) guarantees minimum SINR on both links, (29f)–(29g) enforce lifetime and delay bounds, and (29i) encodes the binary resource-allocation decisions.

F. Deep Sleep Optimizer (DSO) Algorithm

The Deep Sleep Optimizer (DSO) [30] is a bio-inspired metaheuristic that emulates the human sleep–wake cycle to balance exploration and exploitation with low computational overhead. Each agent represents a candidate solution and alternates between *sleep* (local refinement) and *wake* (global exploration) phases, guided by homeostatic pressure $\mathcal{H}(t)$ and circadian rhythm $\mathcal{C}(t)$.

The homeostatic dynamics are modeled as [31]:

$$\mathcal{H}(t) = \begin{cases} \mathcal{H}_o(t)\, e^{(t_o - t)/x_s}, & \text{Sleep phase,} \\ \mu(t) + (\mathcal{H}_o(t) - \mu(t))\, e^{(t_o - t)/x_w}, & \text{Wake phase,} \end{cases} \tag{31}$$

where x_s and x_w are sleep/wake indices, $\mu(t) \in [0, 1]$ is the wake–sleep threshold, and $\mathcal{H}_o(t)$ is the initial homeostatic value. Upper and lower bounds are constrained by

$$\mathcal{H}^{\max}(t) = \mathcal{H}_o^{+} + a\mathcal{C}(t), \quad \mathcal{H}^{\min}(t) = \mathcal{H}_o^{-} + a\mathcal{C}(t), \tag{32}$$

Algorithm 2: Deep Sleep Optimizer (DSO) Algorithm

Input: $\mathcal{N}$: agents, $\mathcal{I}$: iterations, $\mathcal{T}$: circadian cycle, a, x_s, x_w, $\mathcal{H}_o^+$, $\mathcal{H}_o^-$, UB, LB
Output: $\mathcal{F}_{sol}^*$

$\mathcal{F} \leftarrow$ Preallocate fitness vector of agents;
$\mathcal{F}_{sol} \leftarrow$ Preallocate vector of best fitness per iteration;
Compute circadian periodic function $\mathcal{C}(t)$;
Set homeostatic thresholds $\mathcal{H}^{\max}, \mathcal{H}^{\min}$ using (32);
$\gamma \leftarrow$ Initialize positions of agents randomly;
for $n \leftarrow 1$ **to** $|\mathcal{N}|$ **do**
  $\mathcal{F}(n) \leftarrow$ Evaluate fitness of agent n;
end
$\mathcal{F}_{sol}(1) \leftarrow \min(\mathcal{F})$;
for $i \leftarrow 1$ **to** $\mathcal{I}$ **do**
  for $n \leftarrow 1$ **to** $|\mathcal{N}|$ **do**
    $k \leftarrow$ Index of agent with best fitness;
    $X_{\text{best}} \leftarrow$ Current best solution;
    $\mu \leftarrow$ Random asymptote $\in [0, 1]$;
    Bound μ within $\mathcal{H}^{\min}, \mathcal{H}^{\max}$;
    Compute initial homeostatic value using (33);
    Determine sleep/wake state via (31);
    $\mathcal{F}_n \leftarrow$ Evaluate fitness of updated state;
    if $\mathcal{F}_n \leq \mathcal{F}(n)$ **then**
      $\gamma(n, :) \leftarrow \mathcal{H}$;
      $\mathcal{F}(n) \leftarrow \mathcal{F}_n$;
    end
  end
  $\mathcal{F}_{sol}(i+1) \leftarrow \min(\mathcal{F})$;
end
$\mathcal{F}_{sol}^* \leftarrow (\mathcal{F}_{sol})$;
return $\mathcal{F}_{sol}^*$;

with $\mathcal{C}(t) = \sin\left(\frac{2\pi}{\mathcal{T}}(t - \alpha)\right)$ modeling circadian rhythm over a cycle $\mathcal{T}$ and α a random phase shift.

The initialization term is

$$\mathcal{H}_o(t) = \gamma + r \cdot \left(X_{\text{best}} - \mu(t) X_{\text{mean}}\right), \tag{33}$$

where X_{mean} is the population mean, X_{best} the best agent, $r \sim U(0, 1)$, and γ the initial state. The threshold $\mu(t)$ regulates the search bias: $\mu(t) \to 1$ favors exploration, while $\mu(t) \to 0$ emphasizes exploitation.

DSO avoids premature convergence by alternating agent states: during deep sleep, $\mathcal{H}(t)$ decreases for fine exploitation; during wake, it increases for global exploration. Candidate updates follow a greedy selection policy, maintaining diversity while progressively steering the population toward optimal Pareto solutions.

IV. Result and Analysis

The relay selection problem is formulated as a multi-objective mixed-integer nonlinear program (MO-MINLP), which is non-convex and NP-hard due to interference-coupled SINR, logarithmic rate functions, and binary relay variables. Since global optimality is intractable, we evaluate the proposed Deep Sleep Optimizer (DSO) against three representative baselines: Sequential Quadratic Programming (SQP) [32] as a deterministic solver for relaxed instances, Particle Swarm Optimization (PSO) [33] as a widely adopted swarm algorithm, and Grey Wolf Optimizer (GWO) [34] as a recent

Algorithm 3: Monte Carlo Evaluation of SARS with DSO Optimization

Input: N_{MC} realizations; $T_{\max}$ DSO iterations; system parameters (devices $\mathcal{U}$, SCBs $\mathcal{M}$, trust threshold τ, join radius $d_{\max}$, $C_{\max}$, $b_{\min}$, weights $\{\omega_n\}$).
Output: Average performance metrics across N_{MC}.

for $n = 1$ **to** N_{MC} **do**
  Run Mamdani FIS $\to$ defuzzified score Ω (CoG).
  Map $\Omega \mapsto \Gamma_{u,\tilde{u}}(x)$ via (8); assemble $\Gamma(x)$ (9).
  Compute trust vector $\mathbf{T}$ (10) for f_1.
  Select relays as in Alg. 1 $\to \{\mathcal{U}_k\}, \{h_k\}$.
  Compute $R_{u,k,m}$ as in (7) $\forall$u,k,m.
  Evaluate f_1, f_2, f_3, f_4 via (17), (22), (24), (27).
  Apply constraints: rate, SINR, power, latency shown in (29).
  Initialize DSO population over $\{\mathcal{U}_k\}, \{h_k\}$ using Alg. 8b.
  for $t = 1$ **to** $T_{\max}$ **do**
    Evaluate f_1–f_4 for each candidate;
    Update population using DSO sleep–wake operators;
    Maintain and update Pareto set;
  end
  Store selected solution (knee/best compromise).
end
Compute average performance across N_{MC}.
return Mean and variance of f_1, f_2, f_3, f_4 and Pareto set.

metaheuristic balancing exploration and exploitation. These comparisons allow benchmarking DSO against both deterministic and stochastic search strategies.

The multi-objective problem is solved via weighted-sum scalarization, $\sum_{n=1}^{4} \omega_n f_n$ with $\sum_n \omega_n = 1$, enabling explicit trade-offs among social welfare, lifetime, energy efficiency, and latency. While weighted-sum guarantees Pareto-optimality in convex regions, it cannot fully capture non-convex Pareto fronts. To mitigate this, DSO maintains diverse exploration across the solution space, achieving robust allocations even under weight-driven priorities. Compared to dominance-based methods (e.g., NSGA-II), the weighted-sum with DSO offers a simpler yet effective mechanism to adapt to application-specific requirements (e.g., URLLC vs. mMTC), while preserving Pareto efficiency in convex regions and ensuring tractability for dense 6G-IoT networks.

Fig. (3) shows the utility distribution of SQP, PSO, GWO, and the proposed DSO. SQP yields the lowest median utility ($\sim$45) with limited spread, while PSO and GWO achieve moderate performance (medians $\sim$55 and $\sim$52). DSO attains the highest median utility ($\sim$58) and extends the upper bound to $\sim$113, compared to $\sim$105 for PSO, $\sim$98 for GWO, and $\sim$80 for SQP. Although the minimum utilities remain similar across solvers ($\sim$20–25), DSO improves median performance by 28% over SQP and raises the maximum utility by more than 40%.

For comparison, we benchmark the proposed Social-Aware Relay Selection (SARS) scheme against two baselines: Channel-Aware Relay Selection (CARS), which prioritizes relays based on channel quality (e.g., SINR), similar to [35], and Energy-Aware Relay Selection (EARS), which selects

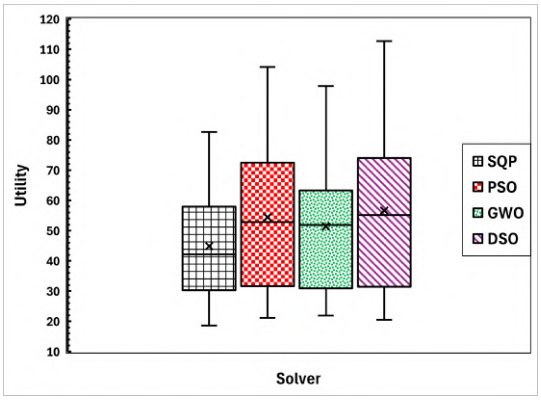


Fig. 3: Utility distribution of SQP, PSO, GWO, and DSO.

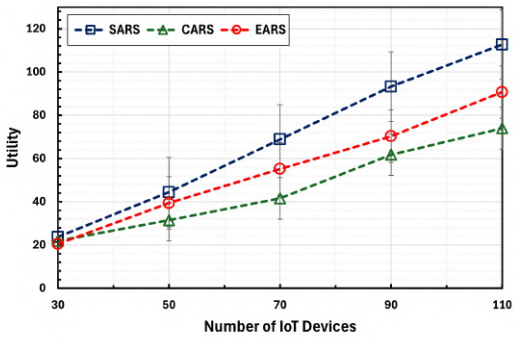


Fig. 4: Utility scalability with increasing number of IoT devices for SARS, CARS, and EARS.

relays according to residual battery levels to maximize network lifetime [36].

Fig. (4) illustrates the scalability of SARS, CARS, and EARS in terms of utility as the number of IoT devices increases from 30 to 110. All schemes show monotonic utility growth, but SARS consistently outperforms, achieving ∼115 at 110 devices compared to ∼95 for EARS and ∼75 for CARS. At lower loads (30 devices), utilities are nearly identical (∼20), yet the performance gap widens with scale: at 70 devices, SARS reaches ∼68, while EARS and CARS attain ∼52 and ∼41, respectively. On average, SARS improves utility by 20–30% over EARS and more than 50% over CARS across all network sizes, confirming its robustness and superior adaptability in dense IoT environments.

As shown in Fig. (5), spectral efficiency increases with the number of IoT devices across all schemes, but the performance gain varies by solver. SARS consistently outperforms CARS and EARS, achieving mean values above 30 bps/Hz under DSO, compared to less than 15 bps/Hz for CARS. DSO yields the highest efficiency across all schemes, surpassing SQP, PSO, and GWO, with improvements of up to 10% over the nearest competitor in SARS and 12% in EARS. These results highlight DSO's robustness in balancing resource allocation under interference-limited and dense deployments,

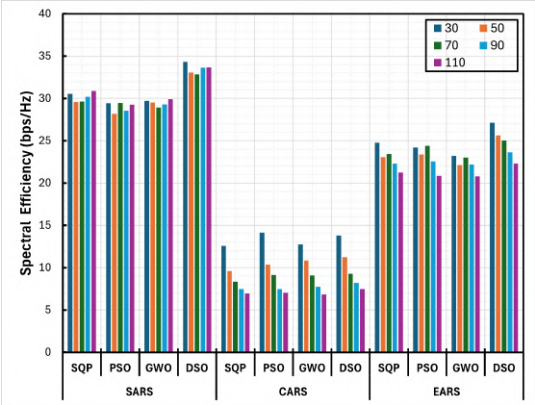


Fig. 5: Mean Spectral efficiency (bps/Hz) of SARS, CARS, and EARS under SQP, PSO, GWO, and DSO for varying IoT device counts

while CARS remains the least efficient due to its dependence on partial CSI.

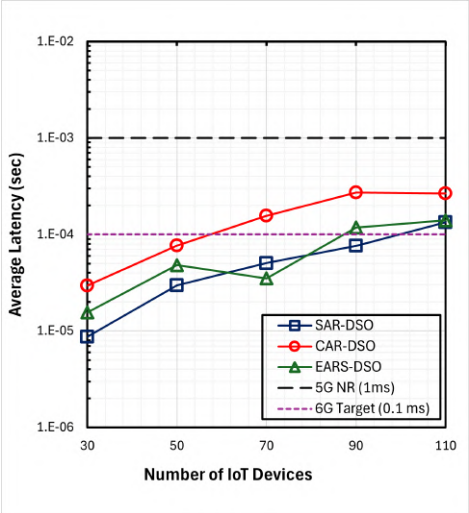


Fig. 6: Average Latency performance of SARS-DSO, CARS-DSO, and EARS-DSO across varying IoT device densities. Results are plotted on a logarithmic scale, with benchmark lines indicating the 5G NR latency requirement (1 ms) and the 6G target (0.1 ms).

The latency performance of SARS, CARS and EARS, using DSO frameworks, is illustrated in Fig. 6. Across all device densities, the measured latency remains well below the 5G NR requirement of 1 ms (black dashed line), confirming compliance with current ultra-reliable low-latency communication (URLLC) standards. Among the schemes, SARS-DSO consistently delivers the lowest latency, maintaining values in the 10 μs to 100 μs range, thereby fully satisfying the 6G target latency of 0.1 ms (100 µs) with substantial margin. EARS-DSO also meets the 6G target across all device densities, although its performance slightly degrades beyond 90 devices.

In contrast, CARS-DSO exhibits higher latency growth, approaching the 6G limit at higher densities ($\sim$110 devices), yet still remains significantly below the 5G NR baseline. Overall, the results demonstrate that SARS-DSO provides the most scalable and robust latency performance, ensuring compliance with both 5G NR and future 6G KPI requirements in dense IoT deployments.

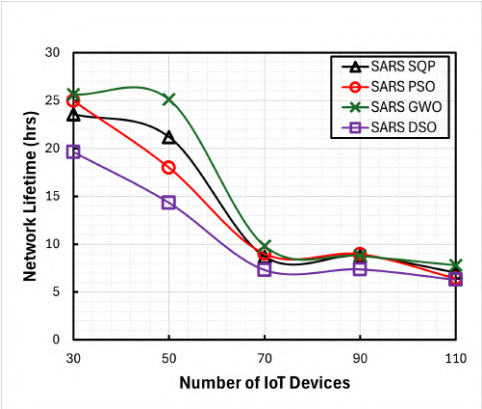


Fig. 7: Network lifetime under SARS with different solvers. Lifetime decreases with network density due to increased energy consumption

Fig. 7 reports the network lifetime (in hours), computed as in (18), where the lifetime of the network is defined as the minimum device lifetime across all participating nodes. A clear density effect is observed: the lifetime decreases monotonically as the number of IoT devices increases from 30 to 110. At low density (30–50 devices), GWO attains the longest lifetime (≈ 26 h with 30 to 50 Iot devices), followed by SQP ($\approx$23–21 h) and PSO ($\approx$24–18 h), while DSO yields the shortest lifetime (≈ 20 h at 30 and ≈ 14 h at 50). Around 70 devices, all methods drop to $\approx$ 7–9 h, and beyond 90 devices the curves largely converge ($\approx$ 6–8 h at 90–110), indicating that energy constraints dominate and solver choice has diminishing impact. The slightly lower lifetime of DSO reflects its balancing of multiple objectives rather than pure energy preservation.

Figs. (8a) and (8b) illustrate the distribution of solutions under SQP and DSO, respectively. In Fig. (8a), SQP converges to a narrow region of the objective space, with most solutions clustering around mid-range values and limited exploration beyond $SW > 0.8$. This reflects the tendency of deterministic solvers to get trapped in local optima, resulting in sub-optimal trade-offs among network lifetime, energy efficiency, and latency. In contrast, Fig. (8b) shows that DSO achieves a wider Pareto spread, with solutions extending across the full range of social welfare and other objectives. The broader distribution indicates superior exploration of the search space, enabling better coverage of trade-offs and higher-quality solutions. These results confirm the advantage of DSO in balancing conflicting objectives while avoiding premature convergence, thereby improving robustness for dense 6G-IoT deployments.

Fig. 9 illustrates the Pareto front approximation under DSO trade-offs between social welfare and the remaining objectives: network lifetime, energy efficiency, and average latency. In Fig. 9a, the Pareto set demonstrates that achieving higher social welfare often comes at the expense of reduced network lifetime, as devices with strong social ties may exhibit faster energy depletion. Conversely, Fig. 9b highlights that while social welfare and energy efficiency are partially aligned, extreme improvements in one metric diminish the other, reflecting the inherent resource constraints in dense IoT networks. Finally, Fig. 9c shows the latency trade-off, where minimizing delay tends to limit achievable social welfare due to stricter scheduling and rate constraints. Overall, the distribution of Pareto-optimal points confirms the ability of the SARS framework to balance competing objectives, enabling flexible operation across diverse application scenarios such as URLLC and mMTC.

V. Conclusion

By integrating weighted-sum scalarization with the Deep Sleep Optimizer (DSO), our framework preserves Pareto efficiency in convex regions while enabling adaptive trade-offs among social welfare, energy sustainability, and latency guarantees in dense 6G-IoT networks. Comparative evaluations against SQP, PSO, and GWO confirm that DSO achieves higher utility, broader Pareto coverage, and improved balance across conflicting objectives. These results highlight the potential of the proposed SARS framework to support reliable and socially aware relay selection in beyond-5G/6G deployments, ensuring both performance and sustainability.

References

[1] S. A. Sarkodie, M. Y. Ahmed, and P. A. Owusu, "Advancing cop26 climate goals: Leveraging energy innovation, governance readiness, and socio-economic factors for enhanced climate resilience and sustainability," *Journal of Cleaner Production*, vol. 431, p. 139757, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095965262303915X

[2] Annu and P. Rajalakshmi, "Towards 6g v2x sidelink: Survey of resource allocation—mathematical formulations, challenges, and proposed solutions," *IEEE Open Journal of Vehicular Technology*, vol. 5, pp. 344–383, 2024.

[3] D. Vukobratovic, M. Ghassemian, X. An, A. F. Coskun, E. Arslan, M. Basaran, N. Bartzoudis, E. Parés, J. Fernandes, L. Cordeiro, L. Rosa, S. Gupta, S. Aguilar, S. Papaharalabos, P. Karlsson, J. Kappi, A. B. Pizarro, M. Aparicio, and A. M. Valenzuela, "A 6g vision for intelligent emergency communication: Unified 3d networking and ambient iot," in *2024 IEEE Globecom Workshops (GC Wkshps)*, 2024, pp. 1–6.

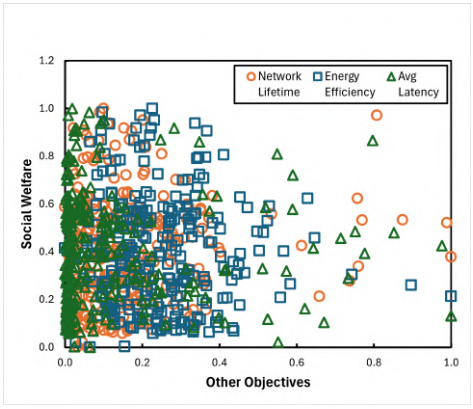
[4] Z. Qadir, K. N. Le, N. Saeed, and H. S. Munawar, "Towards 6g internet of things: Recent advances, use cases, and open challenges," *ICT express*, vol. 9, no. 3, pp. 296–312, 2023.

[5] J. R. Bhat and S. A. Alqahtani, "6g ecosystem: Current status and future perspective," *IEEE Access*, vol. 9, pp. 43 134–43 167, 2021.

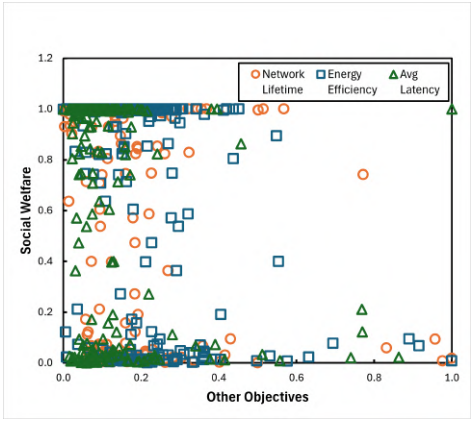
[6] T. Huang, W. Yang, J. Wu, J. Ma, X. Zhang, and D. Zhang, "A survey on green 6g network: Architecture and technologies," *IEEE Access*, vol. 7, pp. 175 758–175 768, 2019.

[7] Z. Lou, B. E. Y. Belmekki, and M.-S. Alouini, "Coverage analysis of hybrid rf/thz networks with best relay selection," *IEEE Communications Letters*, vol. 27, no. 6, pp. 1634–1638, 2023.

[8] D. Kim, M. R. Castellanos, and R. W. Heath, "Joint relay selection and beam management based on deep reinforcement learning for millimeter wave vehicular communication," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 10, pp. 13 067–13 080, 2023.

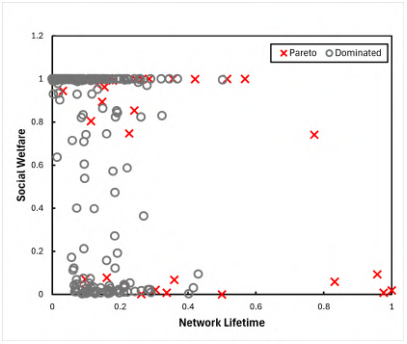


(a) Pareto front approximation under SQP.

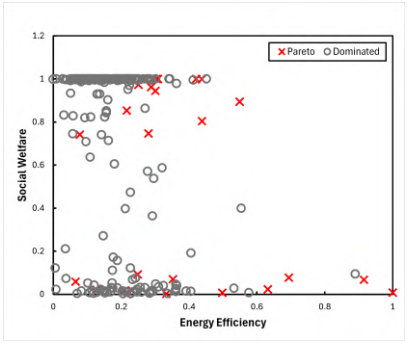


(b) Pareto front approximation under DSO.

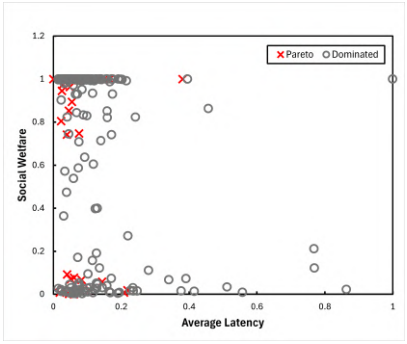
Fig. 8: Pareto front approximations of the proposed multi-objective formulation under (a) SQP and (b) DSO, illustrating the distribution of trade-offs among social welfare, network lifetime, energy efficiency, and latency.



(a) Social welfare vs. network lifetime Pareto front.



(b) Social welfare vs. energy efficiency Pareto front.



(c) Social welfare vs. average latency Pareto front.

Fig. 9: Pareto front visualization of trade-offs among objectives in the proposed SARS framework using DSO.

[9] S. Suyama, M. Iwabuchi, K. Goto, K. Tateishi, and T. Murakami, "Recent activities on new radio network topology for 5g evolution and 6g," in *2024 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*. IEEE, 2024, pp. 1–4.

[10] D. Prabakar, N. Lakshmy, and S. Vishnu, "Dynamic channel state information based relay selection in device-to-device communication," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2020, pp. 744–747.

[11] Y. Bai, D. Wang, G. Huang, and B. Song, "A deep-reinforcement-learning-based social-aware cooperative caching scheme in d2d communication networks," *IEEE Internet of Things Journal*, vol. 10, no. 11, pp. 9634–9645, 2023.

[12] K. Aravind and P. K. R. M, "Optimized fuzzy logic based energy-efficient geographical data routing in internet of things," *IEEE Access*, vol. 12, pp. 18 913–18 930, 2024.

[13] D. K. Sharma, J. Mishra, A. Singh, R. Govil, K. K. Singh, and A. Singh, "Optimized resource allocation in iot using fuzzy logic and bio-inspired algorithms," *Wireless Personal Communications*, vol. 131, no. 2, pp. 1393–1413, 2023.

[14] M. Ayedi, "Enhanced meta-heuristic optimization of resource efficiency in multi-relay underground wireless sensor networks," *PeerJ Computer Science*, vol. 9, p. e1357, 2023.

[15] C. Ramesh Babu and S. Nandakumar, "Towards energy-efficient joint relay selection and resource allocation for d2d communication using hybrid heuristic-based deep learning," *Scientific Reports*, vol. 15, no. 1, p. 25179, 2025.

[16] P. A. Raj, M. Rajakumaran, S. Palani Murugan, and S. Senthilkumar, "Hybridization of metaheuristic algorithms for resource scheduling in distributed robotic control system," *Discover Applied Sciences*, vol. 7, no. 5, pp. 1–21, 2025.

[17] Z. Mohammadi, M. Soleimanpour-Moghadam, S. Talebi, and H. Ahmadi, "Joint network lifetime maximization and relay selection design in underwater acoustic sensor networks," *IEEE Systems Journal*, vol. 18, no. 1, pp. 516–526, 2023.

[18] J. Zhou and S. Tang, "Relay selection for over-the-air computation achieving both long lifetime and high reliability," *Sensors*, vol. 23, no. 8, p. 3824, 2023.

[19] 3rd Generation Partnership Project (3GPP), "Technical specification group services and system aspects; proximity-based services (prose); stage 2," 3GPP, Tech. Rep. TS 23.303, Rel-12, 2014, v12.5.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=729

[20] ——, "Architecture enhancements for 5g system (5gs) to support nr sidelink," 3GPP, Tech. Rep. TS 23.287, 2022, release 17. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3546

[21] ——, "Study on nr v2x sidelink enhancement," 3GPP, Tech. Rep. TR 38.885, 2022, release 17. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3389

[22] ——, "Nr; overall description; stage 2," 3GPP, Tech. Rep. TS 38.300, 2020, release 16. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3191

[23] A. Dogra, R. K. Jha, and S. Jain, "A survey on beyond 5g network with the advent of 6g: Architecture and emerging technologies," *IEEE access*, vol. 9, pp. 67 512–67 547, 2020.
[24] 3rd Generation Partnership Project (3GPP), "NR; Physical channels and modulation (Release 17)," 3GPP, Tech. Rep. TS 38.211, 2024, release 17. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3213
[25] L. A. Zadeh, "Fuzzy logic," *Computer*, vol. 21, no. 4, pp. 83–93, 1988.
[26] S. Kundu, "The min-max composition rule and its superiority over the usual max-min composition rule," *Fuzzy Sets and Systems*, vol. 93, no. 3, pp. 319–329, 1998.
[27] E. Z. Serper and A. Altın-Kayhan, "Coverage and connectivity based lifetime maximization with topology update for wsn in smart grid applications," *Computer Networks*, vol. 209, p. 108940, 2022.
[28] J.-Y. Ji and M. L. Wong, "Decomposition-based multiobjective optimization for nonlinear equation systems with many and infinitely many roots," *Information Sciences*, vol. 610, pp. 605–623, 2022.
[29] Z. Wang, Y. Pei, and J. Li, "A survey on search strategy of evolutionary multi-objective optimization algorithms," *Applied Sciences*, vol. 13, no. 7, p. 4643, 2023.
[30] S. O. Oladejo, S. O. Ekwe, L. A. Akinyemi, and S. A. Mirjalili, "The deep sleep optimizer: A human-based metaheuristic approach," *IEEE Access*, vol. 11, pp. 83 639–83 665, 2023.
[31] A. C. Skeldon, D.-J. Dijk, and G. Derks, "Mathematical models for sleep-wake dynamics: comparison of the two-process model and a mutual inhibition neuronal model," *PloS one*, vol. 9, no. 8, p. e103877, 2014.
[32] C. T. Lawrence and A. L. Tits, "A computationally efficient feasible sequential quadratic programming algorithm," *Siam Journal on optimization*, vol. 11, no. 4, pp. 1092–1118, 2001.
[33] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle swarm optimization: A comprehensive survey," *Ieee Access*, vol. 10, pp. 10 031–10 061, 2022.
[34] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.
[35] M. Ghorbanzadeh and A. Abdelhadi, *Practical Channel-Aware Resource Allocation*. Springer, 2022.
[36] J. Li, R. Chai, C. Liu, C. Liang, Q. Chen, and F. R. Yu, "Energy-aware joint route selection and resource allocation in heterogeneous satellite networks," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 8, pp. 12 067–12 081, 2024.

Estimating QoE of Encrypted Video Streams, a Machine Learning Approach

1st Best Nkhumeleni
Department of Electrical Engineering
University of Cape Town
Cape Town, South Africa
nkhbes001@myuct.ac.za

2nd Joyce Mwangama
Department of Electrical Engineering
University of Cape Town
Cape Town, South Africa
joyce.mwangama@uct.ac.za

***Abstract*—With video traffic comprising the majority of Internet usage, monitoring user Quality of Experience (QoE) has become a critical challenge, particularly as end-to-end encryption renders deep packet inspection ineffective. This paper presents the design and implementation of a machine learning–based QoE prediction framework that operates solely on packet-level statistics extracted from encrypted video streams. The framework captures packet traces, extracts statistical features such as bitrate, packet size, and inter-arrival times, and employs trained Random Forest classifiers to infer resolution in real-time, a key video quality indicator. Evaluation on diverse video traces demonstrates that the system achieves high accuracy in predicting resolution and produces QoE estimates closely aligned with the ground truth table, producing a model with an 85.7% accuracy. The proposed framework enables operators and researchers to monitor QoE in encrypted environments without access to payload data, offering a scalable and privacy-preserving approach to video quality monitoring.**

***Index Terms*—QoE, live prediction, QoS, packet-level statistics, Random forest, chunk detection**

I. Introduction

Video traffic has become the dominant component of global Internet usage, driven largely by adaptive streaming services such as YouTube and Netflix. Ensuring that end users receive a high Quality of Experience (QoE) is therefore critical for content providers, Internet service providers, and researchers. However, the widespread adoption of end-to-end encryption has rendered traditional monitoring approaches, such as deep packet inspection, ineffective, leaving operators with limited visibility into video playback performance. This creates a pressing need for methods to infer QoE without relying on access to video payloads [1], [2].

QoE estimation is inherently challenging because it depends on multiple interrelated factors, including video resolution, startup delay, stalling events, and network conditions. Among these, video resolution serves as one of the most visible and directly perceived indicators of user experience. Prior research has demonstrated the potential of statistical analysis and machine learning to derive QoE metrics from encrypted traffic patterns. For example, systems like Requet leverage chunk-level features to predict buffer events and playback quality in real time. While these efforts highlight the feasibility of payload-agnostic QoE monitoring, there remains a need for frameworks that can be flexibly applied using packet-level statistics alone [1].

In this work, we present a framework for real-time QoE prediction that focuses specifically on video resolution estimation from encrypted traffic. Our approach captures packet traces within a controlled environment, extracts packet-level features such as bitrate, packet count, inter-packet intervals, and mean packet size, and applies supervised machine learning models to predict the resolution of the video stream.

This study is guided by the following research questions:

1) **RQ1:** How can Quality of Experience (QoE) for encrypted video streams be estimated without access to the underlying video bitstream?
2) **RQ2:** Which machine learning models (e.g., Random Forest, SVM, Logistic Regression) are most effective at predicting QoE-related metrics such as resolution from packet-level Quality of Service (QoS) features?
3) **RQ3:** Is live prediction of QoE possible with access to only packet level statistics?
4) **RQ4:** How accurate can a model be, the predicts QoE from live collected traces?

The rest of this paper is organised into the following sections. Section 2: reviews existing work on QoE and Quality of Service (QoS), discusses the challenges posed by encrypted streaming, and explores major approaches such as man-in-the-middle proxies, parametric models (e.g., ITU-T P.1203), and machine learning techniques. The use of synthetic data for class balancing and model training is also examined. Section 3 describes the framework used for this research, including dataset preparation, feature extraction from packet captures, machine learning model design, and evaluation strategies. Details on the experimental setup and performance metrics are also provided. Section 4 presents the outcomes of the experiments, compares model performance, and discusses the effectiveness of different approaches in predicting QoE from encrypted video streams. The advantages and limitations of each model are critically analyzed. Finally section 5 summarizes the main findings of the research, reflects on its contributions, and outlines potential directions for future studies, including improving generalization across networks, reducing computational complexity, and validating synthetic

data generation methods.

II. Literature Review

QoE estimation for encrypted HTTP Adaptive Streaming (HAS) services (e.g., YouTube, Netflix) is challenging because HTTPS hides the bitstream, pushing researchers toward indirect, traffic-based inference [3]–[5]. While QoS provides measurable network metrics (latency, jitter, throughput, bitrate), QoE reflects end-user perception and must be inferred or modeled [6].

A. Approach Families

Two dominant families appear in the literature:

1) **Man-in-the-Middle (MitM) proxy**: Intercept/decrypt to access detailed session events (startup delay, stalls, bitrate), enabling rich ground truth at the cost of intrusiveness and possible QoS offsets [5]. MitM frameworks provide fine-grained labels but may perturb measurements as they can introduce an offset to QoE metrics and lack scalability [5].
2) **Application-agnostic inference**: Use parametric QoE models (e.g., ITU-T P.1203) and/or supervised ML on encrypted-flow features to estimate QoE without decryption, trading intrusiveness for dependence on representative data and features [2], [7]. 5G/DASH ML studies map network QoS features (RTT, throughput, packet counts) to QoE/KQI targets across hybrid, buffer-based, and rate-based ABR strategies, often without explicit chunk detection, achieving $\sim$80–90% accuracy in static/mobility scenarios [8], [9]. In-network monitoring combines short-interval KPI estimation with the validated P.1203 model for session-level and near-real-time QoE as implemented by [2], [7].

Commonly used Machine Learning models for encrypted traffic include Random Forests for robust classification on tabular, noisy features [10]; SVM / Logistic Regression when decision boundaries are simpler or approximately linear [11], [12]; and KNN, Decision Trees, Gradient Boosting, Naive Bayes applied contextually; boosting often excels with careful regularization [13]–[17].

B. Trade-offs and Takeaways

- Intrusiveness vs. scalability: MitM yields precise labels but is operationally heavy; encrypted-traffic ML and parametric models scale better yet rely on data quality [2], [5], [7].
- Model selection: For complex, non-linear QoS$\rightarrow$QoE mappings, ensembles (RF/GB) are frequently preferred; for simpler relations, SVM/linear models suffice [10]–[12], [15].
- Data strategy: High-quality ground truth remains pivotal [3], [4]. Synthetic augmentation can stabilize training under imbalance but must be validated against real distributions [18], [19].

In summary, state-of-the-art QoE estimation for encrypted streaming balances deployment practicality with prediction fidelity. Non-intrusive, ML-driven methods—often anchored by Random Forests or boosting and supported by ITU-T P.1203—achieve competitive accuracy while preserving encryption [7], [10], [15]. Future work should enhance cross-network generalization, reduce feature-engineering overhead, and systematize synthetic-data validation to minimize real-world performance drift [19], [20].

III. Methodology

This section describes the process of designing, implementing, and evaluating the proposed Quality of Experience (QoE) predictor. The framework consists of four main stages: (i) packet trace collection, (ii) feature extraction, (iii) machine learning model training and evaluation and (iv) live qoe implementation. The predicted resolution is used as the primary QoE indicator.

A. *Packet Trace Collection*

The Requet dataset [1] was used as the foundation for packet trace collection. This dataset provides PCAP files of video sessions along with corresponding log files that contain playback and buffering metadata. Leveraging these resources allowed for both network-layer and application-layer perspectives of the same video stream.

To integrate these traces into the QoE prediction framework, a custom pipeline was developed. The process began by cloning the RequetDataSet repository and iterating through each experimental group. This was all automated in the `requet_end_to_end.py` for every video session, the following steps were carried out:

1) **Parsing Playback Metadata:** The `merged.txt` logs were parsed into structured data frames, extracting playback quality states, buffer health, and progress information for each video stream, from that information the resolution/video quality of the video stream is ascertained and used to form the ground truth table.
2) **Packet Loading and Flow Grouping:** Corresponding PCAP files were read using Scapy. Each packet was mapped to its transport flow, with uplink (client $\rightarrow$ server) and downlink (server $\rightarrow$ client) directions separated. This enabled precise identification of request–response cycles.
3) **Chunk Detection:** GET requests in the uplink were used as triggers for chunk boundaries. Downlink traffic exceeding thresholds was aggregated as video data. For each chunk, features such as time-to-first-byte (TTFB), chunk duration, slack time, download time, and chunk size were computed.
4) **Dataset Construction:** The final output was a nested list of video chunks annotated with their associated resolution, serialized for downstream machine learning models.

This multi-stage pipeline ensured that traces were both reproducible and aligned with the QoE prediction task. By combining playback logs with packet-level captures, it was

possible to reliably label each chunk with its resolution and derive the statistical features required for supervised learning.

B. Feature Extraction

Once packet traces were collected and aligned with playback logs, the next step involved extracting descriptive features from the captured video traffic. The extraction was performed at the granularity of *chunks*, where a chunk corresponds to a continuous segment of video data transfer initiated by an uplink request (e.g., an HTTP GET) and ending before the next request. Each chunk is annotated with the ground-truth resolution obtained from the playback metadata, forming the input–label pair for supervised learning.

The following features were derived for every detected chunk:

1) **Chunk Detection:** The boundary of a chunk is identified when the client issues a sufficiently large uplink request to the video server (typically on TCP port 443). The first downlink response that surpasses a threshold marks the start of data delivery. All packets until the next GET request are aggregated as part of the same chunk. This procedure ensures that each chunk corresponds to a meaningful video segment.
2) **Time-to-First-Byte (TTFB):** TTFB is defined as the time difference between the GET request that initiates a chunk and the arrival of the first downlink packet. It captures network responsiveness and initial buffering delay, which strongly impact perceived start-up latency and stall events.
3) **Chunk Duration:** The total time span from the first packet of a chunk until the last packet received. Chunk duration indicates how long it takes to deliver an entire segment of video data and reflects throughput stability.
4) **Slack Time:** Slack time is the idle period between the end of one chunk and the next uplink request. It is computed as the difference between the timestamp of the last packet of the current chunk and the first packet of the following GET request. Large slack times typically correspond to buffered playback, while small slack times may indicate rebuffering risks.
5) **Download Time:** The effective time required to transfer the chunk payload, calculated as the chunk duration minus slack and TTFB. This value reflects how efficiently the available bandwidth is utilized for video delivery.
6) **Chunk Size:** The total number of bytes transferred in downlink packets within the chunk. Larger chunk sizes are generally associated with higher-quality segments (e.g., 1080p vs. 360p) and thus provide discriminative information for resolution prediction.

By combining these temporal and size-based features, each chunk is represented as a structured vector that captures both transport-level efficiency (e.g., TTFB, download time) and video quality indicators (e.g., chunk size, duration). These feature vectors form the dataset used to train machine learning models to predict the video resolution and, by extension, the user's Quality of Experience (QoE).

C. Model Training and Evaluation

After constructing the chunk-level feature set, we trained a random forest model to predict the video *resolution label* per chunk. The full pipeline closely follows the code listing provided and proceeds as follows.

a) Data assembly and labels.: Two serialized datasets were loaded and concatenated: `all_chunks_no1080.pkl` and `all_chunks_1080.pkl`, yielding a single list of chunk records. For each chunk, the feature vector is

$$\mathbf{x} = [\text{time_stamp},\ \text{chunk_duration},\ \text{ttfb},\ \text{slacktime},\ \text{download_time},\ \text{chunk_size},\ \text{audio_video}]$$

and the categorical target $y \in \{360\text{p}, 480\text{p}, 720\text{p}, 1080\text{p}\}$ (or the subset present) is the ground-truth resolution derived from playback logs. Labels were mapped to integers with a `LabelEncoder` to obtain y_{enc}.

b) Train/validation split.: We performed an 80/20 hold-out split with stratification by class to preserve the empirical class distribution:

$$(\mathbf{X}_{\text{train}}, \mathbf{X}_{\text{test}}, y_{\text{train}}, y_{\text{test}}) \sim \text{StratifiedSplit}(\mathbf{X}, y_{\text{enc}}, 0.8),$$

using a fixed random seed (`random_state=42`) for exact reproducibility.

c) Learning algorithm and hyperparameters.: A Random Forest (RF) classifier was trained on $\mathbf{X}_{\text{train}}$ with `n_estimators=100` and `random_state=42`. The RF was selected for its robustness to non-linear decision boundaries and mixed-scale features, and its ability to expose feature importances.

Hyperparameter	Value
Estimator	`RandomForestClassifier`
`n_estimators`	100
`criterion`	Gini (default)
`max_depth`	None (nodes expand until pure)
`class_weight`	None (implicit via stratified split)
`random_state`	42

d) Evaluation on hold-out.: Predictions $\hat{y}$ were generated for $\mathbf{X}_{\text{test}}$. We report:

- **Overall accuracy** $\text{Acc} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{\hat{y}_i = y_i\}$.
- **Confusion matrix** $C \in \mathbb{N}^{K\times K}$ with $C_{r,c}$ = #(true class $= r$, predicted $= c$), where rows are ground truth and columns are predictions (as printed in the code).
- **Per-class precision/recall/F1**

e) Cross-validation.: To assess stability beyond a single split, we computed 5-fold cross-validated accuracy on the full dataset using `cross_val_score`:

$$\text{CV@5} = \frac{1}{5}\sum_{j=1}^{5} \text{Acc}\big(\mathcal{D}_{\text{val}}^{(j)}\big),$$

reporting the mean $\pm$ standard deviation across folds.

f) Feature importance.: Post-training, the RF's impurity-based importances ϕ_d were extracted and ranked, providing model-internal evidence of which features are most discriminative for resolution. In our pipeline the seven features are: {*time_stamp, chunk_duration, ttfb, slacktime, download_time, chunk_size, audio_video*}.

g) Pseudocode (as implemented).:

1: Build feature matrix $\mathbf{X} \in \mathbb{R}^{N\times 7}$ from first 7 fields of each chunk; targets y from field 8 (resolution).
2: Encode labels: $y_{\text{enc}} \leftarrow$ `LabelEncoder.fit_transform`(y); store class order `classes_`.
3: Stratified split: 80% train / 20% test with `random_state=42`.
4: Train RF: `RandomForestClassifier`(100, `random_state=42`).`fit`($\mathbf{X}_{\text{train}}, y_{\text{train}}$).
5: Predict: $\hat{y} \leftarrow$ `predict`($\mathbf{X}_{\text{test}}$).
6: Report metrics: accuracy, confusion matrix (rows=true, cols=pred), classification report (per-class P/R/F1).
7: 5-fold CV: `cross_val_score`(RF, $\mathbf{X}, y_{\text{enc}}$, cv $= 5$) $\rightarrow$ mean $\pm$ std accuracy.
8: Feature importance: sort `feature_importances_` descending and display.

h) Notes on validity and reproducibility.: (i) *Stratification* mitigates class imbalance in the hold-out. (ii) *Fixed seeds* (`random_state=42`) ensure deterministic splits and model fits. (iii) The 5-fold CV offers a distribution of accuracies that complements the single hold-out figure. (iv) Because chunks may originate from the same session, future work could incorporate a *grouped* split (e.g., by video/session) to prevent potential leakage across folds if session-level correlations are strong. (v) results may vary depending on the network conditions used when streaming.

D. *Online Chunking & Live Resolution Inference*

Beyond offline parsing of PCAP files, a live pipeline was implemented to (i) capture encrypted video traffic in real time, (ii) detect chunk boundaries on the fly, (iii) compute the same seven features used in training, and (iv) infer per-chunk resolution using the trained Random Forest and label encoder. The implementation leverages `Scapy`'s packet sniffer and a lightweight state machine to maintain timing and size counters with sub-second latency.

a) Capture setup.: Given a network interface `iface` (default `wlp4s0`), the client IPv4 address is obtained programmatically and used to disambiguate flow directions. Packets are filtered with a BPF filter `tcp port 443` to restrict processing to HTTPS flows.

For each packet, the code extracts $(t, \ell, \text{src}, \text{dst}, s_p, d_p)$ where ℓ is the TCP payload length (not the full frame). This matches the online design goal of minimizing per-packet overhead while retaining transport-level signal.

b) State and thresholds.: The detector maintains a small set of scalar state variables: stream start time t_0, current chunk start t_{start}, slack marker t_{slack}, accumulated downlink bytes $S_{\downarrow}$, time-to-first-byte T_{TTFB}, and two booleans to gate transitions (`we_got_first_uplink`, `first_downlink`). Per-flow uplink/downlink payload buffers are kept in dictionaries (`uplink`, `downlink`). Three thresholds govern chunking and feature validity:

$$\text{GET_THRESH} = 300\ \text{B}, \quad \text{DOWN_THRESH} = 300\ \text{B}$$

$$\text{MIN_CHUNK_SIZE} = 600\ \text{B}.$$

Uplink accumulation $>$ GET_THRESH fires a new chunk boundary; the first downlink accumulation $>$ DOWN_THRESH inside a chunk latches T_{TTFB}. Chunks are only emitted (and hence classified) if $S_{\downarrow} \geq$ MIN_CHUNK_SIZE to suppress spurious micro-bursts.

c) Feature computation (online).: At the instant an uplink trigger fires, the seven features are emitted for the *previous* chunk window:

$$\mathbf{x} = [\ \underbrace{t - t_0}_{\text{time_stamp}}\ ,\ \underbrace{t - t_{\text{start}}}_{\text{chunk_duration}}\ ,\ T_{\text{TTFB}},\ \underbrace{t - t_{\text{slack}}}_{\text{slacktime}}],$$

$$\underbrace{(t - t_{\text{start}}) - (t - t_{\text{slack}}) - T_{\text{TTFB}}}_{\text{download_time}},\ \underbrace{S_{\downarrow}}_{\text{chunk_size}}\ ]$$

This mirrors the offline extractor so the live model consumes identically structured inputs.

d) Inference and visualization.: If $S_{\downarrow} \geq$ MIN_CHUNK_SIZE, the trained RF model (`rf_qoe_model.pkl`) predicts a resolution code, which is mapped back to a string label (e.g., `720p`) via `rf_label_encoder.pkl`. For interpretability on a live dashboard, the numeric component (e.g., 720) is plotted against stream time. A short window mode filter

$$\tilde{r}(t) = \text{mode}\left(r(t-w+1), \ldots, r(t)\right), \quad w = 3,$$

is applied to stabilize oscillations from transient misclassifications. The plot updates in real time using Matplotlib's interactive mode.

e) State transitions and resets.: On each emitted chunk, the detector *fully resets* per-flow buffers and timing flags, and sets the new chunk origin to the current timestamp. This enforces strict segmentation and prevents cross-chunk leakage:

$$S_{\downarrow} \leftarrow \text{uplink_bytes}, \quad t_{\text{start}} \leftarrow t, \quad t_{\text{slack}} \leftarrow t, \quad T_{\text{TTFB}} \leftarrow 0,$$

$$\texttt{first_downlink} \leftarrow \text{False}.$$

Separately, every downlink packet updates t_{slack}; only the first downlink meeting DOWN_THRESH latches T_{TTFB}.

f) Pseudocode of the live loop.:

Initialize $t_0, t_{\text{start}}, t_{\text{slack}}, S_\downarrow, T_{\text{TTFB}} \leftarrow 0$; flags $\leftarrow$ `False`; buffers $\leftarrow \varnothing$
for each packet p from `sniff` on TCP:443 **do**
  if p not (IPv4 & TCP) **then**
    continue
  end if
  Extract $(t, \ell, \text{src}, \text{dst}, s_p, d_p)$, compute $t_{\text{rel}} = t - t_0$
  if t_0 unset **then**
    $t_0 \leftarrow t$; $t_{\text{start}} \leftarrow t$
  end if
  if src=client && $d_p = 443$ **then** {uplink}
    Append ℓ to uplink buffer for flow, compute $\Sigma_\uparrow$
    if not `we_got_first_uplink` && $\Sigma_\uparrow >$ GET_THRESH **then**
      Compute features (duration, T_{TTFB}, slack, download_time, $S_\downarrow$)
      if $S_\downarrow \geq$ MIN_CHUNK_SIZE **then**
        Predict resolution with RF, plot smoothed mode
      end if
      Reset buffers/flags; set $S_\downarrow \leftarrow \Sigma_\uparrow$; $t_{\text{start}} \leftarrow t$; $t_{\text{slack}} \leftarrow t$; $T_{\text{TTFB}} \leftarrow 0$; `first_downlink`$\leftarrow$False
      `we_got_first_uplink`$\leftarrow$True
    end if
  else if dst=client && $s_p = 443$ **then** {downlink}
    `we_got_first_uplink`$\leftarrow$False
    Append ℓ to downlink buffer for flow, compute $\Sigma_\downarrow$
    if not `first_downlink` && $\Sigma_\downarrow >$ DOWN_THRESH **then**
      $S_\downarrow \leftarrow S_\downarrow + \Sigma_\downarrow$; $T_{\text{TTFB}} \leftarrow t - t_{\text{start}}$; `first_downlink`$\leftarrow$True
    end if
    $t_{\text{slack}} \leftarrow t$
  end if
end for

g) Design considerations and caveats.: (i) **Payload vs. frame length:** live extraction uses TCP payload length (lighter CPU footprint) whereas some offline variants may count full packet bytes; ensure consistency when mixing datasets. (ii) **Threshold calibration:** GET_THRESH, DOWN_THRESH, MIN_CHUNK_SIZE were chosen conservatively for HTTPS video; they can be tuned per CDN/player. (iii) **Label stability:** the w=3 mode filter reduces flicker without masking sustained shifts. (iv) **Session changes:** resets on each detected GET guarantee strict per-chunk independence but may miss segments if an application reuses connections with atypical request patterns.

IV. Results and Experimentation

This section presents the experimental setup, evaluation methodology, and results obtained from training and testing the proposed QoE prediction framework. Both offline (trace-driven) and online (live inference) evaluations are reported.

A. Experimental Setup

- **Datasets:** Packet traces and playback metadata were drawn from the Requet dataset [1], supplemented by live captures generated in a controlled testbed (see Section III-D).
- **Hardware/Software:** All experiments were conducted on i5-10300Hx8, 16GiB ram running Ubuntu 24.04.2 as shown in Figure 1 Models were implemented in Python 3.10 with Scikit-learn, and packet parsing used Scapy.
- **Model configuration:** A Random Forest classifier with 100 estimators and fixed random seed (42) was trained on chunk-level features described in Section 3.

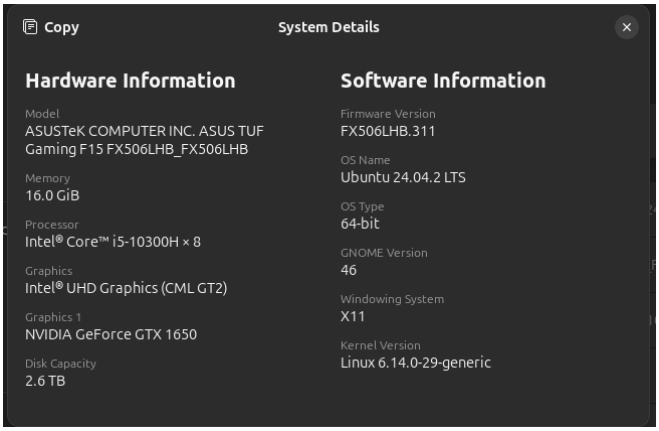


Fig. 1. Computer Hardware info

B. Offline Evaluation

1) Train/Test Split: The dataset was divided using an 80/20 stratified split to preserve class balance across resolution labels. Labels were encoded using a `LabelEncoder`.

2) Accuracy and Confusion Matrix: The model achieved an overall test-set accuracy of **85.7%**. Table I shows the confusion matrix, where rows correspond to true classes and columns to predictions.

TABLE I
Confusion Matrix of Resolution Prediction

	240p	360p	480p	720p	1080p
240p	77	28	50	323	173
360p	12	336	175	1220	274
480p	382	166	382	1141	244
720p	22	164	211	42244	931
1080p	10	106	117	1935	4566

3) Cross-Validation: To test robustness, 5-fold cross-validation was performed. The average accuracy across folds was **75.57%** with a standard deviation of **9.37%**, indicating variable performance .

C. Online Evaluation

The live chunk detector (Section III-D) was tested under real-world streaming conditions. By attaching the sniffer to

the client interface and applying the trained Random Forest classifier, per-chunk resolution predictions were produced in real time and visualized dynamically. To reduce transient misclassifications, a running mode filter with a window size of $w = 3$ was applied to the predicted labels.

*1) **Fixed-Resolution Streams**:* We first evaluated streams locked at a constant resolution (e.g., YouTube videos forced to 480p, 720p, or 1080p). Figures 2–4 show time-series predictions for these cases. The predicted resolution closely follows the ground truth throughout the session, with only minor fluctuations around chunk boundaries.

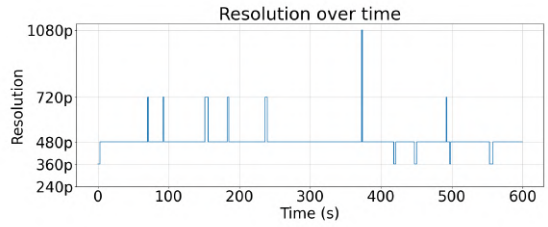


Fig. 2. Live resolution prediction for a fixed 480p stream.

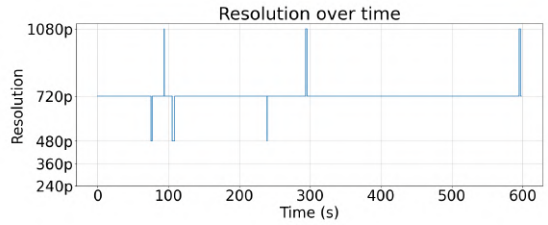


Fig. 3. Live resolution prediction for a fixed 720p stream.

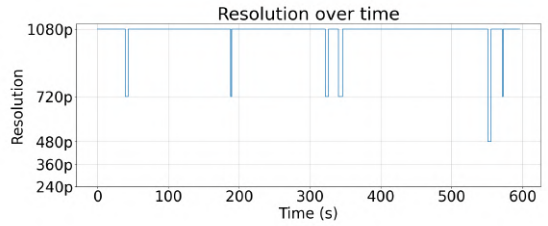


Fig. 4. Live resolution prediction for a fixed 1080p stream.

*2) **Adaptive Bitrate (ABR) Streams**:* To test robustness in dynamic conditions, we streamed videos with adaptive bitrate enabled. Under varying network bandwidth, the player automatically switched between resolutions (e.g., 720p $\rightarrow$ 480p $\rightarrow$ 1080p). The model successfully tracked these transitions, as illustrated in Fig. 5. Predicted resolution changes closely aligned with the actual adaptation events visible in the playback metadata.

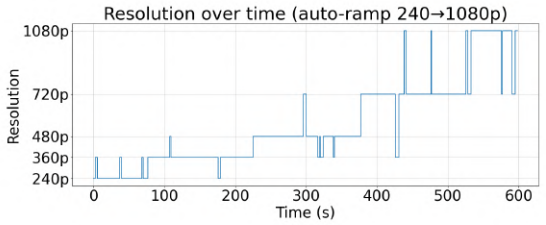


Fig. 5. Live resolution prediction during an adaptive stream with multiple quality shifts.

*3) **Short Clips and Variable Length Streams**:* We further evaluated the detector on short video clips ($<$ 2 minutes) and long-form streams ($>$ 10 minutes). In shorter clips, the model required fewer chunks before stabilizing, while in longer sessions it continued to produce stable predictions over extended durations. This highlights the ability of the system to operate continuously without manual resets.

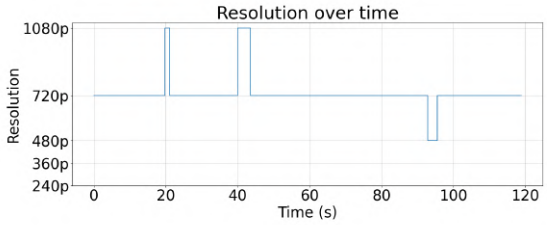


Fig. 6. Live resolution for a 720p stream over 2 minutes

*4) **Summary of Online Evaluation**:* These experiments confirm that the live system:

- Accurately predicts fixed-resolution streams (360p, 480p, 720p, 1080p).
- Correctly follows adaptive bitrate switching in real time.
- Operates reliably across both short and long streaming sessions.

The results demonstrate the feasibility of real-time encrypted stream monitoring for QoE prediction without payload inspection.

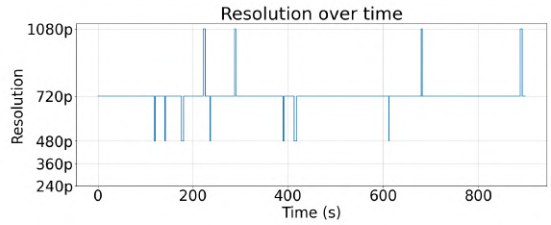


Fig. 7. Live resolution for a 720p stream over 15 minutes

V. Conclusion

This work demonstrated that packet-level statistics, without any payload inspection, are sufficient to infer a key proxy for video Quality of Experience (QoE) in encrypted environments. We designed a reproducible pipeline that (i) aligns PCAP traces with playback logs, (ii) extracts robust chunk-level features, and (iii) trains a supervised model to predict video resolution in real time. On a stratified hold-out set the Random Forest achieved an overall accuracy of **85.7%**, while 5-fold cross-validation yielded **75.57%** $\pm$ **9.37%**. Feature analysis highlighted `chunk_size` and `download_time` as the most discriminative predictors, which is consistent with intuition about bitrate- and throughput-driven adaptive streaming. A live sniffer validated that the model tracks both fixed-resolution sessions and adaptive bitrate (ABR) shifts with low latency, confirming practical feasibility for online monitoring.

Limitations. Our current evaluation may reflect session-level correlations within the dataset; grouped splits by session/video would further guard against leakage. Thresholds used for online chunking were tuned conservatively for HTTPS video and may require calibration across CDNs, players, or transport protocols (e.g., QUIC/HTTP/3). Finally, we use *resolution* as the primary QoE indicator; while salient, QoE also depends on startup delay and rebuffering.

Future work. We plan to (i) adopt group-aware validation and cross-dataset tests for stronger generalization claims, (ii) incorporate additional QoE targets (startup delay, stall ratio) via multi-task learning, (iii) explore domain adaptation to new networks and players, (iv) harden the online chunker for QUIC/HTTP/3 and mixed traffic, and (v) deploy lightweight models on edge devices for scalable, privacy-preserving operations.

In summary, this framework provides a practical path to QoE visibility under end-to-end encryption, enabling operators and researchers to monitor user-perceived quality using only transport-layer signals while preserving user privacy.

VI. Acknowledgments

This research is further supported by; Telkom South Africa, via the Telkom Centre of Excellence Programme; Telkom-NRF through the Future Technologies Programme; and Sentech through their Centres of Excellence Programme.

References

[1] C. Gutterman, K. Guo, S. Arora, X. Wang, L. Wu, E. Katz-Bassett, and G. Zussman, "Requet: Real-time qoe detection for encrypted youtube traffic," in *Proceedings of the 10th ACM Multimedia Systems Conference*, 2019, pp. 48–59.

[2] I. Orsolic and L. Skorin-Kapov, "A framework for in-network qoe monitoring of encrypted video streaming," *IEEE access*, vol. 8, pp. 74 691–74 706, 2020.

[3] M. V. D. Merwe, "Web-application traffic classification using machine learning in a software-defined network," Masters. dissertation, University of CapeTown, 2019.

[4] S. Louw, "Qos and/or qoe prediction of encrypted video internet traffic," Honours. thesis, University of CapeTown, 2021.

[5] S. Göring, A. Raake, and B. Feiten, "A framework for qoe analysis of encrypted video streams," in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, 2017, pp. 1–3.

[6] M. Varela, L. Skorin-Kapov, and T. Ebrahimi, "Quality of service versus quality of experience," in *Quality of Experience: Advanced Concepts, Applications and Methods*. Springer, 2014, pp. 85–96.

[7] W. Robitza, S. Göring, A. Raake, D. Lindegren, G. Heikkilä, J. Gustafsson, P. List, B. Feiten, U. Wüstenhagen, M.-N. Garcia *et al.*, "Http adaptive streaming qoe estimation with itu-t rec. p. 1203: Open databases and software," in *Proceedings of the 9th ACM Multimedia Systems Conference*, 2018, pp. 466–471.

[8] R. Ul Mustafa, D. Moura, and C. E. Rothenberg, "Machine learning approach to estimate video qoe of encrypted dash traffic in 5g networks," in *2021 IEEE Statistical Signal Processing Workshop (SSP)*, 2021, pp. 586–589.

[9] R. Ul Mustafa, S. Ferlin, C. Esteve Rothenberg, D. Raca, and J. J. Quinlan, "A supervised machine learning approach for dash video qoe prediction in 5g networks," in *Proceedings of the 16th ACM symposium on QoS and security for wireless and mobile networks*, 2020, pp. 57–64.

[10] S. J. Rigatti, "Random forest," *Journal of Insurance Medicine*, vol. 47, no. 1, pp. 31–39, 2017.

[11] H. Wang and D. Hu, "Comparison of svm and ls-svm for regression," in *2005 International conference on neural networks and brain*, vol. 1. IEEE, 2005, pp. 279–283.

[12] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of biomedical informatics*, vol. 35, no. 5-6, pp. 352–359, 2002.

[13] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[14] Y.-Y. Song and L. Ying, "Decision tree methods: applications for classification and prediction," *Shanghai archives of psychiatry*, vol. 27, no. 2, p. 130, 2015.

[15] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in neurorobotics*, vol. 7, p. 21, 2013.

[16] V. Nasteski, "An overview of the supervised machine learning methods," *Horizons. b*, vol. 4, no. 51-62, p. 56, 2017.

[17] G. I. Webb, E. Keogh, and R. Miikkulainen, "Naïve bayes." *Encyclopedia of machine learning*, vol. 15, no. 1, pp. 713–714, 2010.

[18] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[19] T. E. Raghunathan, "Synthetic data," *Annual review of statistics and its application*, vol. 8, no. 1, pp. 129–140, 2021.

[20] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018.

Security, Privacy, and Trust

Investigating the Robustness of Audio Features Under Concept Drift in Android Malware Detection

1st Mussa Phiri
Department of Computer Science
University of Zululand
KwaDlangezwa, South Africa
202073198@stu.unizulu.ac.za

2nd Paul Tarwireyi
Department of Computer Science
University of Zululand
KwaDlangezwa, South Africa
tarwireyip@unizulu.ac.za

3rd Alfredo Terzoli
Department of Computer Science
University of Zululand
KwaDlangezwa, South Africa
terzolia@unizulu.ac.za

***Abstract*—Android malware detection is a continuous challenge, as attackers frequently change their techniques and the Android ecosystem evolves. Machine learning models trained on historical data often degrade in performance due to concept drift, where the statistical properties of features change over time. Although most research has examined permissions and API calls, little is known about the stability of audio-based features. In this study, we investigate the temporal stability of audio features in Android applications. Using three drift measures, Population Stability Index (PSI), Jensen–Shannon distance (JS), and Kolmogorov-Smirnov statistic (KS), we evaluate feature families in quarterly malware bins. Results show that drift is widespread but uneven: features such as Zero-Crossing Rate (ZCR) and Chroma exhibit high instability, while spectral descriptors such as Spectral Flatness remain stable. A drift gap analysis confirms that the instability increases with the temporal distance. Finally, negative Spearman correlations between drift metrics and weighted F1-scores of non-incrementally trained models show that drift contributes directly to performance degradation. These findings establish that audio features are affected by concept drift and provide a baseline for future work on adaptive and multimodal detection strategies.**

***Index Terms*—Concept Drift, Android Malware Detection, Population Stability Index, Jensen–Shannon distance, Kolmogorov-Smirnov Test, Spearman Correlation.**

I. Introduction

Smartphones have become indispensable in daily life, with Android remaining the dominant operating system worldwide. This is primarily due to its open-source nature and high customizability. The operating system currently accounts for over 80% of the global mobile market [1]. However, this popularity makes Android a prime target for cybercriminals. According to [1], Android accounts for most mobile malware attacks, registering 98% of mobile attacks. Attackers continuously evolve their tactics to evade detection, and to address this, machine learning–based malware detection has gained traction, often relying on static and dynamic features such as API calls, permissions, and bytecode instructions [2] [3]. Recently, alternative feature representations have emerged, including images derived from binaries [3] and audio-based transformations of Android Package Files (APK) [4]. The conversion of APK files to audio allows for the extraction of audio features, including chroma, spectral, and cepstral features. Beyond offering a fresh perspective, this approach is effective in handling obfuscated or encrypted binaries, as it can capture underlying patterns or anomalies that might be missed when using binary format [5]. Furthermore, the lightweight nature of audio processing makes it suitable for large-scale analysis and near-real-time applications.

Although these approaches have demonstrated promising classification performance, their long-term reliability has not yet been studied. This study seeks to investigate the robustness of Android audio-based features to concept drift. Concept drift refers to changes in the statistical properties of data over time, which was shown to lead to performance degradation of machine learning models [6] in textual-based features. In the cybersecurity field, drift is especially prevalent as malware constantly evolves due to the continuous adaptation of tactics, techniques, and obfuscation strategies. Android, in particular, has an ecosystem that constantly evolves through regular OS updates and shifting development practices, while adversaries adapt malware strategies to evade detection.

Drift can be formally described using probability distributions over the model inputs and outputs. We denote the input features as X , Y as the true labels, and y the predictions of the model. *Concept drift* occurs when the conditional distribution $p(Y \mid X)$ changes over time, reflecting a change in the relationship between features and ground-truth labels. Other forms of drift include *prediction drift*, which is the change in $p(y \mid X)$, and *label drift*, which is the change in the distribution $p(Y)$. *Feature drift* refers to the change in the distribution of input data $p(X)$ [7].

In order to differentiate the effects of concept drift, two cases are considered. *Virtual drift* arises when $p(X)$ changes, but $p(Y \mid X)$ remains stable, which means that the feature distribution changes without impacting the predictive accuracy of machine learning models. In contrast, *real drift* occurs when $p(Y \mid X)$ itself changes, directly leading to degraded model performance. These distinctions are well established for traditional Android malware features, such as API calls and permissions. However, it is not yet known whether audio-based features are equally vulnerable to such drifts or whether certain families exhibit inherent resilience. Figure 1 shows an illustration of virtual drift versus real drift.

This paper addresses that gap by performing a statistical investigation of concept drift in audio-based features for Android malware detection. Using established drift metrics, including

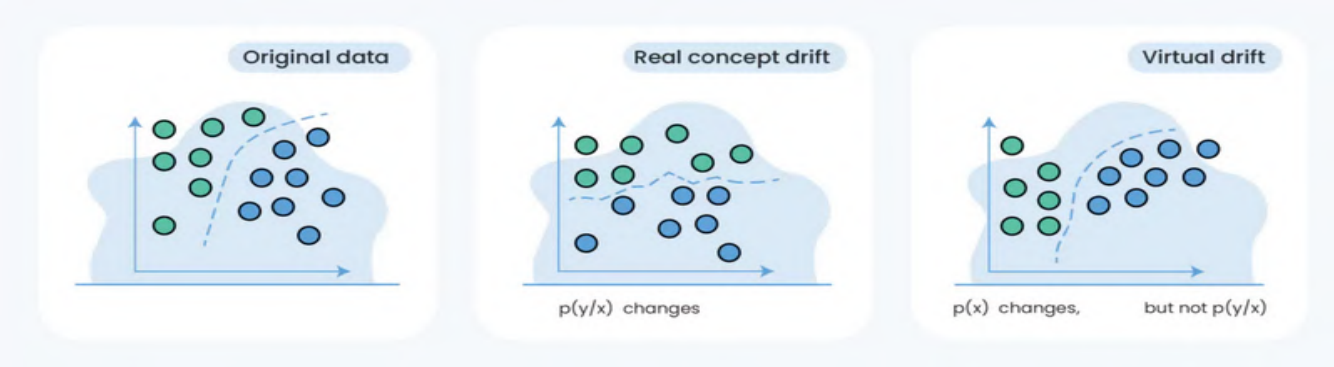


Fig. 1. Virtual Drift vs Real Drift. Source: [7]

the Population Stability Index (PSI), Kolmogorov–Smirnov (KS) test, and Jensen–Shannon (JS) distance, we measure distributional changes across a time-binned malware dataset. We further provide feature family stability rankings and correlations between drift severity and model degradation in non-incremental settings.

II. Related Work

Concept drift is a critical challenge in machine learning, arising when models trained on historical data degrade as underlying patterns evolve in new data. This section reviews prior work in two parts: general advances in drift detection and studies focused on Android malware.

A. Concept Drift in Machine Learning

The authors in [8] reviewed concept drift detection strategies across machine learning domains. They noted the common use of statistical tests (PSI, KL, JS, KS) and error-rate based detectors (DDM, EDDM, ADWIN). Ensemble and neural approaches were seen as effective but costly, while statistical and window-based methods were faster for streaming data. The study concluded that no single method dominates and highlighted gaps in non-tabular domains, such as image and audio.

The authors [9] developed CDSeer, a semi-supervised, model-agnostic drift detection framework designed for real-world deployment. Unlike prior methods such as ECHO, which required extensive manual labelling, CDSeer reduced labelling to less than 1% of data through clustering-based sampling and pseudo-labelling. It was compatible with pre-trained models and adaptable across architectures. Evaluation on eight datasets showed CDSeer achieved higher precision and recall than ECHO while matching the performance of fully supervised detectors, providing a practical solution for managing drift in deployed ML systems.

The work by [10] introduced a model-based explanation framework for concept drift using explainable AI (XAI) methods. Instead of only detecting or quantifying drift, their approach localized and segmented drift regions and then applied explanation techniques such as global feature importance, discriminative dimensionality reduction, local explanations (LIME, saliency maps), and counterfactuals. Experiments on synthetic datasets, Bayesian networks, sensor data, and images (MNIST, Fashion-MNIST, ImageNet) demonstrated the ability to highlight drifting features and visualize how drift manifests. The study underscored the importance of interpretable explanations to increase trust and improve decision-making in non-stationary environments.

B. Concept Drift in Android Malware Detection

While general machine learning research has established a wide range of techniques for detecting and managing concept drift, domain-specific studies are essential to understand how these methods perform under adversarial and evolving environments. Android malware detection is a particularly challenging domain, as attackers frequently adapt their strategies, and the Android ecosystem itself evolves rapidly. Recent works have begun to apply drift-aware methods to sustain classifier performance in this setting, primarily focusing on static features such as API calls and permissions.

[11] proposed a data stream learning pipeline that adapts both the classifier and feature extractor in response to drift, using detectors such as ADWIN and KSWIN to trigger retraining. Unlike methods that update only classifiers, refreshing feature extractors reduced feature drift and improved detection. On DREBIN and AndroZoo, ADWIN with retraining achieved F1-scores of 84.44% and 75.05%, outperforming standard approaches and stabilizing performance over time.

[12] performed a comprehensive drift analysis across 12 Android malware detectors based on string, image, and graph features. To quantify distributional change, they applied the Kolmogorov–Smirnov (K–S) test, which revealed significant shifts in feature distributions, particularly in function call graph structures under obfuscation. Their experiments showed that image-based detectors degraded the fastest, losing up to 38% F1-score, while graph-based models were more resilient but weak against unseen families. A key finding was that models trained on recent samples performed poorly on older data, underscoring the "forgetting effect" and reinforcing the need for adaptive or drift-aware methods.

[13] extended the Transcend framework into TRANSCENDENT, a rejection-based strategy for handling concept drift

in malware detection. The study used Kullback–Leibler (KL) divergence to show that prior evaluations exaggerated drift (KL = 696.66) by merging datasets, while their temporal AndroZoo dataset reflected more realistic evolution (average KL = 189.55). For drift detection, TRANSCENDENT relied on conformal evaluation with p-values, rejecting uncertain samples to prevent misclassification. Using data from 2014–2019, the approach maintained F1-scores above 0.8 for nearly two years, doubling classifier lifespan while keeping rejected samples almost entirely those that would have been errors.

[14] studied Android permissions for drift-resilient detection using the KronoDroid dataset (2008–2020, 79K apps). A dynamic Random Forest ensemble with time-based updates sustained an average F1 of 0.93 over seven years. The Kolmogorov–Smirnov test showed no significant difference between 60 and 122 permissions, confirming the smaller set's stability. Performance dips in 2015–2016 were linked to evolving ransomware families, showing that drift-aware permission models can remain effective long-term.

C. Audio-Based Android Malware Detection

Besides permissions and API calls, recent studies have explored audio-based representations of APK files for Android malware detection. These approaches treat bytecode as a signal and extract audio features, including spectral, cepstral, and chroma features.

The authors of [15] proposed an Android malware detection method using sonified audio features. Android APK files were converted to audio signals, from which 191 static audio-based features were extracted. The study utilised nature-inspired wrapper-based metaheuristic algorithms, including the Genetic Algorithm, to optimise feature selection. The best results were achieved with the Genetic Algorithm, which reduced the feature set by 50.26% and achieved a classification accuracy of 99.72% using the Light Gradient Boosting Machine (LGBM) model.

The authors in [16] proposed an Android malware detection method utilizing audio features. The authors converted .dex files from the Android package to audio files. They extracted features including chromagrams, spectral contrast, and Root Mean Square. They also utilized feature selection methods like information gain and gain ratio, which helped reduce computational costs. The authors achieved the best accuracy score of 96.6% with the KNN and ReliefF feature selection method.

The authors in [4] proposed an Android malware detection method using audio feature fusion. Android Application Package (APK) components were converted to binary and then to audio. Bark frequency Cepstral Coefficients (BFCC) features, and Gammatone Frequency Cepstral Coefficients (GFCC) features were then extracted from the CICMalAnal2017 and CICMalDroid2020 datasets. The study achieved the best classification accuracy of 98.96% using the Light Gradient Boost Machine (LGBM) model.

While these studies demonstrate the effectiveness of audio features for malware detection, they assume a stationary feature space and do not assess robustness under temporal or distributional drift. Evaluating this limitation is crucial for establishing the long-term reliability of audio-based techniques.

III. Research Method

The primary objective of this study is to evaluate the robustness of audio features to concept drift in Android malware detection. The methodology consists of three main stages: (i) transforming Android applications into audio signals and extracting feature families, (ii) quantifying temporal drift across quarterly dataset bins using multiple statistical measures, and (iii) validating the impact of drift by correlating drift magnitudes with classification performance in a non-incremental setting. Figure 2 illustrates the overall research methodology followed in this study.

A. Dataset Description

The study uses the AndroOBFS dataset [17], which contains 16,279 unique real-world obfuscated Android malware samples. The dataset only contains malware samples which are categorised into six obfuscation categories: (i) *Trivial*, (ii) *Renaming*, (iii) *Encryption*, (iv) *Reflection*, (v) *Code*, and (vi) *Mix* (a combination of two or more obfuscation methods). The dataset spans the period 2018–2020 and is partitioned into quarterly bins, allowing for temporal analysis experiments. Table I shows the quarterly distribution of the dataset.

TABLE I
AndroOBFS Distribution per Quarter

Quarter	Code	Encryption	Mix	Reflection	Renaming	Trivial
2018_Q1	613	301	80	66	140	46
2018_Q2	708	330	70	64	181	51
2018_Q3	788	390	81	88	175	53
2018_Q4	1158	548	143	123	236	81
2019_Q1	860	407	138	84	194	59
2019_Q2	707	320	109	67	163	47
2019_Q3	1160	525	205	118	270	80
2019_Q4	1341	651	250	137	328	93
2020_Q1	69	48	7	7	29	10
2020_Q2	368	178	60	43	100	29
2020_Q3	133	76	9	17	35	10
2020_Q4	86	59	13	12	32	11

B. Feature Extraction

The Android Application Package (APK) comprises multiple files required for the execution of the Android application, including classes.dex, the manifest, and the resource files. These files differ across applications, making each APK unique. However, the raw package files cannot be directly used in machine learning models. These files were first transformed into binary and then concatenated into a one-dimensional binary array, which was converted into a waveform representation (WAV). From the resulting audio signals, a diverse set of audio descriptors was extracted, including cepstral (MFCCs), spectral (centroid, bandwidth, contrast, flatness, roll-off), temporal (ZCR, RMS), and chroma (STFT, CENS, CQT) features, as well as Mel spectrograms and polynomial coefficients.

- **Mel-frequency cepstral coefficients** – These are audio features based on human perception of loudness and

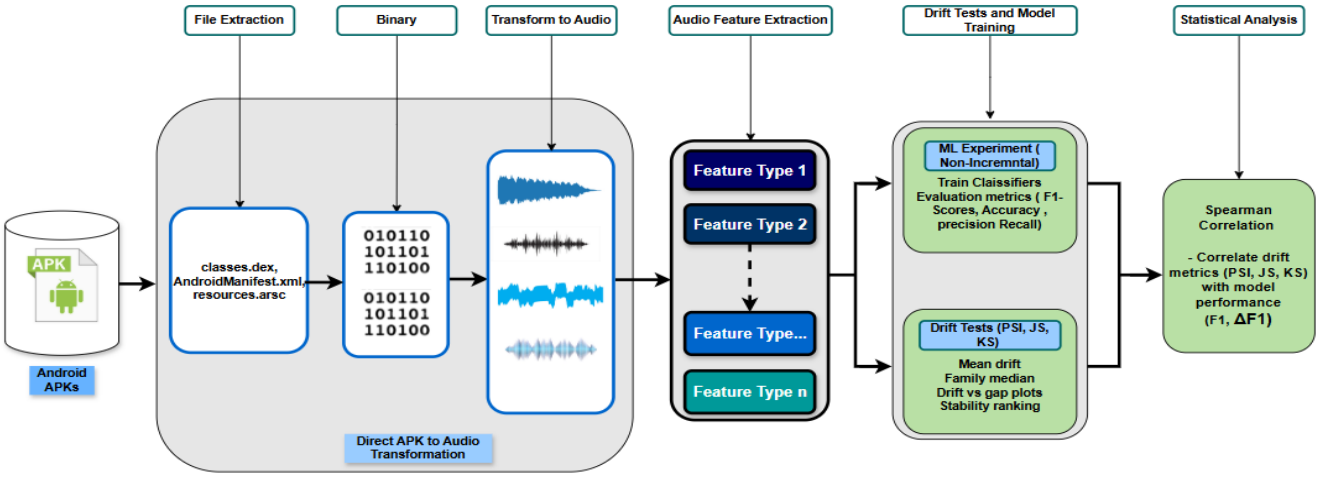


Fig. 2. Methodology. Adapted from [15]

frequency of sound using the Mel scale. The implementation process involves windowing the audio signal, applying the Discrete Fourier Transform (DFT), taking the logarithm of the signal size, warping the frequencies on a Mel scale, and applying the inverse Discrete Cosine Transform (DCT) [4].

$$f_{\text{mel}} = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right), \tag{1}$$

where f_{mel} is the frequency on the Mel scale, and f is the frequency in Hz.

$$f = 700 \cdot \left(10^{\frac{f_{\text{mel}}}{2595}} - 1\right), \tag{2}$$

- **Zero Crossing Rate** – Measures how often the signal changes sign or crosses zero per unit time. It is a basic feature often used in speech and music classification [4].
- **Spectral Centroid** –This describes where the centre of the frequency content is located in a sound. It is found by calculating the weighted average of all the frequencies in the signal. It is linked to the perceived brightness of the sound [15]. The equation is shown below:

$$C_i = \frac{\sum_{k=1}^{W_f L} k \cdot X_i(k)}{\sum_{k=1}^{W_f L} X_i(k)} \tag{3}$$

- **Spectral Contrast** – Captures the decibel difference between peaks and troughs in the frequency spectrum, distinguishing harmonic content. High contrast usually means that the sound is clear, while low contrast may suggest that it is noisier. [16].
- **Spectral Bandwidth** – Measures the width of the frequency spectrum by quantifying the spread of frequencies around the spectral centroid. A wider bandwidth indicates more high-frequency content, while a narrower one suggests a more tonal or low-frequency-dominated signal [16].
- **Spectral Flatness** – Quantifies how noise-like or tone-like a signal is by comparing the geometric mean to the arithmetic mean of the power spectrum. A value close to 1 indicates a flat, noise-like spectrum (white noise), while values near 0 correspond to tonal signals with distinct peaks [16].
- **Spectral Rolloff** –This is the frequency below which a certain percentage, usually 85% or 95% of the total spectral energy is contained [18].

$$\sum_{k=1}^{m} X_i(k) = C \cdot \sum_{k=1}^{W_f L} X_i(k) \tag{4}$$

where C is the adopted percentage (user-defined parameter). The spectral rolloff frequency is usually normalized by dividing it by $W_f L$, so that its value lies between 0 and 1.

- **Root Mean Square (RMS)** – RMS (Root Mean Square) is used to estimate how powerful or loud a sound is within a short time window. It calculates the average energy of the audio signal and gives an indication of its overall intensity [16].
- **Chroma STFT** – Derived from the Short-Time Fourier Transform (STFT), this feature projects the audio spectrum into twelve semitone bins (one for each pitch class of the octave). It captures harmonic and melodic content by tracking how pitch class energy evolves over time [15] [19].
- **Chroma CENS** – Chroma Energy Normalized Statistics (CENS) are computed by applying additional energy normalization and statistical smoothing to the chromagram. This makes the feature more robust to dynamics and noise while retaining pitch-class distribution [20].

- **Chroma CQT** – Computed using the Constant-Q Transform (CQT), this chroma variant provides higher frequency resolution at lower frequencies and lower resolution at higher frequencies. It preserves harmonic structure by mapping energy into twelve pitch classes [20].
- **Mel-Spectrogram** – A time–frequency representation where the frequency axis is mapped to the perceptually motivated mel scale. Unlike MFCCs, which compress spectral shape into cepstral coefficients, the mel-spectrogram preserves detailed temporal and spectral patterns, making it useful for distinguishing subtle audio characteristics [16] [20].
- **Poly Features** – These features fit an n-th order polynomial to the power spectrum of each frame, returning the polynomial coefficients. They capture the overall spectral trend and curvature, serving as a compact descriptor of spectral shape [16].

C. Statistical Drift Tests

To capture distributional changes, drift was computed between consecutive quarterly bins of the AndroOBFS dataset (2018–2020).In order to quantify temporal instability in audio features, three statistical drift measures were used, including the Population Stability Index (PSI), the Kolmogorov–Smirnov (KS) test, and the Jensen–Shannon divergence (JS). Drift was first computed at the *coefficient level* for each feature family. For multi-dimensional features such as MFCC and Chroma, every coefficient was expanded and evaluated individually across consecutive quarterly bins. Family-level statistics were then obtained by aggregating these coefficient scores, reporting both the median and mean values per bin-pair. This two-level approach allowed for both fine-grained analysis of coefficient behaviour and broader comparisons across entire feature families.

1) Population Stability Index (PSI): The Population Stability Index (PSI) measures the difference between a given distribution and a reference distribution, such as datasets from different quarters or years. A PSI below 0.1 indicates stability, 0.1 to 0.25 suggests moderate change, and values above 0.25 signify significant drift [21].

$$\text{PSI} = \sum_i (p_i - q_i) \times \ln\left(\frac{p_i}{q_i}\right) \tag{5}$$

where p_i and q_i are the proportions of the i-th bin in the new and baseline distributions, respectively.

2) Kolmogorov–Smirnov (KS) Test: The Kolmogorov–Smirnov test is a non-parametric method used to assess whether two distributions differ significantly. It calculates the maximum distance between the empirical cumulative distribution functions (ECDFs) of two samples [22]:

$$D = \sup_x |F_n(x) - G_m(x)| \tag{6}$$

where $F_n(x)$ and $G_m(x)$ are the ECDFs of the two samples, and $\sup$ is the supremum (maximum) across all x.

3) Jensen–Shannon (JS) Divergence: The Jensen–Shannon divergence is a symmetric and smoothed version of Kullback–Leibler divergence, used to compare two probability distributions P and Q (e.g., feature histograms). It is defined as:

$$JSD(P \parallel Q) = \tfrac{1}{2} D_{KL}(P \parallel M) + \tfrac{1}{2} D_{KL}(Q \parallel M) \tag{7}$$

where $M = \frac{1}{2}(P+Q)$ is the average distribution, and D_{KL} is the Kullback–Leibler divergence. In this study, we report the Jensen–Shannon distance (JS), which is the square root of the JS divergence after base-2 normalisation. JS Distance is symmetric and bounded in [0, 1] (0 = identical, 1 = maximally different), which makes it easier to interpret. [21].

D. Model Training and Evaluation

Finally, to link statistical drift to practical consequences, non-incremental classifiers were trained to predict obfuscation categories using a 70/30 split of each quarterly bin. The models were trained in the training set for a given quarter and then evaluated on the test data held from the same quarter and all subsequent quarters. For classification, three machine learning models were employed: a linear *Passive Aggressive Classifier*, *Multinomial Naive Bayes*, and a tree-based *Gradient Boosting Classifier*. This setup simulates a static deployment scenario and allows one to quantify how performance degrades when models are not adapted to evolving data distributions. The performance of the models was evaluated using various metrics, including accuracy, precision, recall, and F1-score. Due to the class imbalance of the dataset, a weighted F1-score was used, as it accounts for the imbalance by averaging the F1 of each class according to its support. This was also the primary metric for correlation analysis between drift and classifier performance.

$$\text{F1}_{\text{weighted}} = \sum_{c \in C} \frac{n_c}{N} \cdot \frac{2 \times \text{Precision}_c \times \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}, \tag{8}$$

where n_c is the number of samples in class c, N is the total number of samples, and C is the set of classes.

E. Statistical Analysis

To examine the relationship between feature drift and model performance, Spearman's rank correlation was used. Spearman correlation is a non-parametric measure that evaluates the strength and direction of the monotonic association between two variables [23]. In this study, it was applied to assess the degree of correlation between drift metrics (PSI, JS, KS) and the weighted F1-scores of the classifiers. Given a sample size n and rank differences d_i between paired observations, the Spearman correlation coefficient ρ is defined as:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}. \tag{9}$$

A negative correlation indicates that higher drift values are associated with lower classification performance, thereby directly linking feature instability to model degradation.

F. Implementation Details

The code for the audio transformation, feature extraction, drift analysis, and classification pipelines was conducted in Python. For audio processing and feature extraction, the soundfile and Librosa (v0.9.2) libraries were used. The drift tests were implemented using NumPy, pandas, and scipy.stats (ks_2samp for the Kolmogorov–Smirnov test). Spearman rank correlations were computed using scipy.stats.spearmanr. Additional preprocessing utilities made use of re and pandas. Graphs were generated using Matplotlib and Seaborn.

Model training and evaluation used scikit-learn, which was also used to compute performance metrics such as accuracy and weighted F1-score. All development and analysis were performed in Jupyter Notebook on a Dell PC running Windows 11 Pro (version 24H2), equipped with a 12th Gen Intel(R) Core(TM) i9-12900 CPU @ 2.40 GHz, 32 GB of RAM, and a 64-bit operating system.

IV. Results

This section presents the experimental results evaluating the robustness of audio features under concept drift in Android malware detection. Drift was assessed using three statistical measures including Population Stability Index (PSI), Jensen–Shannon Distance (JS), and Kolmogorov–Smirnov statistic (KS), across consecutive temporal bins. The study also analysed family-level drift rankings, stability bucket distributions, and drift–time gap relationships. Finally, Spearman correlations were computed between drift scores and non-incremental classifier performance (F1-weighted), using both linear (Passive Aggressive) and tree-based (Gradient Boosting) models.

A. Per-feature drift tests (PSI, JSD, KS)

Table II presents the aggregated drift statistics. The most unstable families were ZCR and Chroma_STFT, with median PSI values above 0.45 and drift detected in over 60% of transitions. Spectral Centroid and RMS also showed high susceptibility to drift, with median PSI values exceeding 0.32. These findings were further supported by KS test results.

Core audio features such as MFCCs and Mel-Spectrograms displayed intermediate stability (median PSI 0.22–0.23; drift prevalence 45–48%), indicating that they were neither the most drift-prone nor the most resilient.

Features including Spectral Flatness, Spectral Bandwidth, and Spectral Rolloff remained largely stable, as they recorded very low median PSI values (< 0.14) and drift prevalence below 20%. Across all features, Jensen–Shannon Distance (JS) values for were moderate (≈ 0.27–0.29 for the most unstable families), while stable families remained lower ($\approx 0.12 - 0.14$), indicating that while the distributions diverged across consecutive bins, their overall shapes were partially preserved. Kolmogorov-Smirnov (KS) statistics were consistent with the PSI rankings, with unstable families recording higher medians (> 0.22) than stable families (~ 0.11–0.13). Overall, PSI, JS, and KS converged on the same stable and drift-prone feature families, reinforcing the robustness of these results.

TABLE II
Ranking of Feature Families by Drift Severity Across Statistical Metrics

FeatureFamily	mean_PSI	median_PSI	mean_JS	median_JS	mean_KS	median_KS	pct_moderate	pct_drift
ZCR	0.473	0.517	0.260	0.292	0.235	0.265	18.18	72.73
Chroma_STFT	0.527	0.467	0.246	0.284	0.238	0.271	21.97	63.64
Spectral_Centroid	0.459	0.419	0.255	0.270	0.220	0.229	9.09	63.64
RMS	0.299	0.326	0.213	0.240	0.148	0.154	18.18	63.64
Poly_Features	0.287	0.259	0.205	0.214	0.175	0.157	22.73	50.00
Chroma_CQT	0.280	0.250	0.201	0.210	0.183	0.182	20.45	50.00
Mel_Spectrogram	0.304	0.234	0.205	0.202	0.174	0.149	24.72	48.37
MFCC	0.280	0.218	0.200	0.195	0.168	0.151	27.95	45.45
Chroma_CENS	0.269	0.191	0.188	0.183	0.167	0.153	30.30	38.64
Spectral_Contrast	0.210	0.142	0.173	0.159	0.135	0.119	29.87	32.47
Spectral_Rolloff	0.168	0.139	0.159	0.157	0.128	0.125	45.45	18.18
Spectral_Bandwidth	0.169	0.109	0.154	0.139	0.115	0.107	54.55	9.09
Spectral_Flatness	0.133	0.091	0.138	0.126	0.114	0.135	36.36	9.09

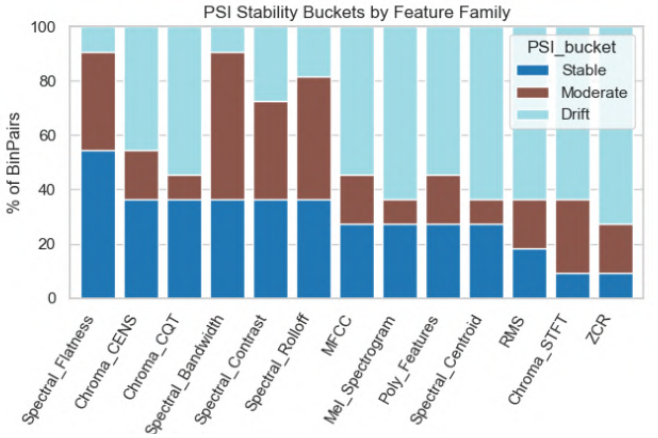


Fig. 3. Stability_bucket

Figure 3 illustrates the distribution of stability outcomes (Stable, Moderate, Drift) across feature families. Consistent with Table I, ZCR and Chroma_STFT show the highest instability, drifting in more than 60% of temporal transitions. Spectral Centroid and RMS also exhibit substantial drift proportions, whereas Spectral Flatness and Spectral Bandwidth remain largely stable, exceeding 80% stability across bins.

Figure 4 tracks median PSI values over time for selected families. The results highlight clear divergence between drift-prone and stable features: ZCR and Chroma_STFT frequently surpass the drift threshold (0.25), peaking above 0.8 in certain quarters, while Spectral Flatness and Spectral Bandwidth remain consistently below the moderate-change threshold (0.1). This temporal view reinforces the conclusion that drift affects feature families unevenly, with some descriptors remaining resilient, while others degrade rapidly.

B. Drift vs Time Gap

Figures 5, 6, and 7 plot drift values against the temporal gap between bins for two drift-prone families (ZCR, Chroma_STFT) and one stable family (Spectral Flatness). The results show a clear accumulation effect across all three metrics (PSI, JS, KS). In PSI, ZCR, and Chroma STFT cross the 0.10 band early (gaps $\approx$ 2–3) and exceed 0.25 by gaps $\approx 3 - 4$, approaching the upper display cap at larger gaps, whereas Spectral Flatness remains below 0.25 until long gaps. For JSD and KS, the horizontal lines are reference guides

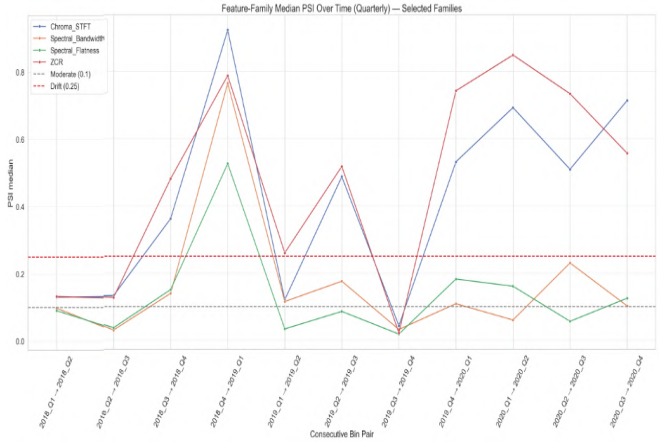


Fig. 4. Temporal PSI feature median

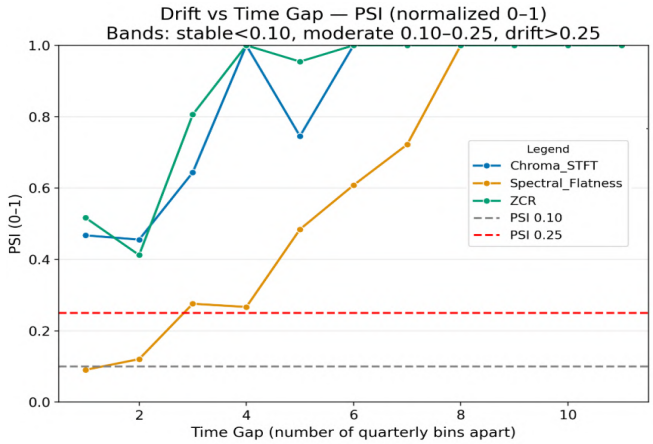


Fig. 5. Drift vs Time Gap (PSI)

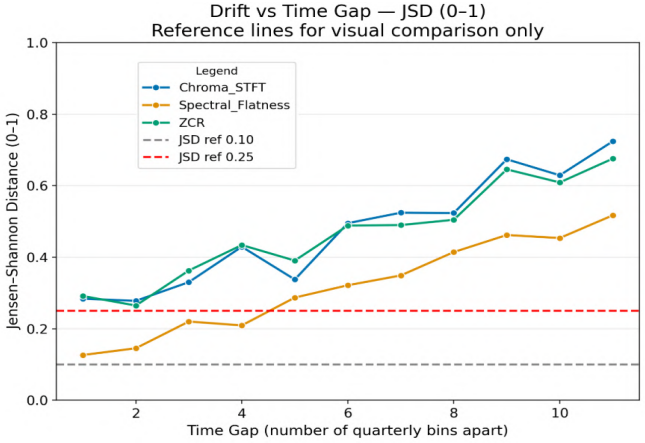


Fig. 6. Drift vs Time Gap (JS)

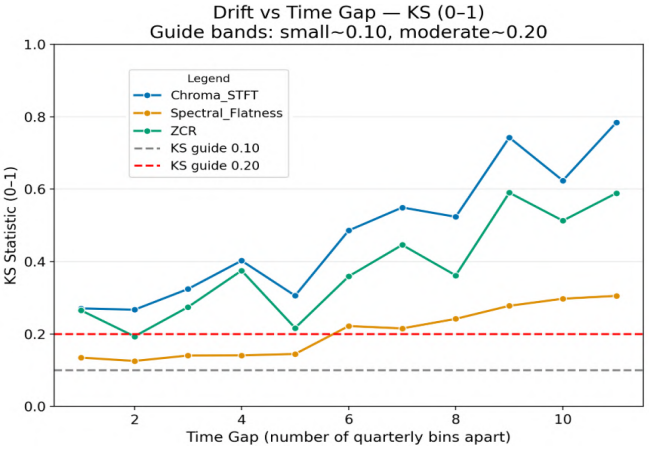


Fig. 7. Drift vs Time Gap (KS)

(not thresholds); the same upward trend is observed, with peaks typically around 0.6–0.7 (JS) and 0.5–0.8 (KS) for drift-prone families, while Spectral Flatness remains lower ($\approx$ 0.2–0.35).These findings complement Figures 3 and 4 by confirming that not only do some families drift more frequently, but their instability also intensifies with increasing temporal separation, underscoring the cumulative nature of concept drift.

C. Feature Stability Rankings

Figures 8, 9 and 10 rank the feature families by their median drift scores, providing a direct comparison of relative stability. The ordering is consistent with earlier findings: Spectral Flatness, Spectral Bandwidth, and Spectral Rolloff emerge as the most stable descriptors, while ZCR and Chroma_STFT are the least stable. Intermediate families such as MFCC, Chroma CQT, and Mel Spectrogram fall in the mid-range, showing partial resilience but still prone to moderate drift. By consolidating PSI, JS, and KS into a composite view, this ranking reinforces the divergence between highly drift-prone temporal features and more stable spectral descriptors.

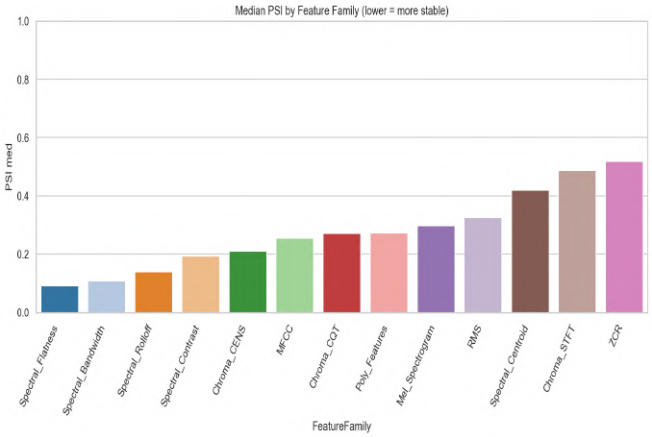


Fig. 8. Feature Family Median PSI

D. Correlation with Model Performance (Spearman)

Table III presents the Spearman correlations between drift metrics and non-incremental F1-weighted performance. All models exhibited strong negative relationships, confirming that

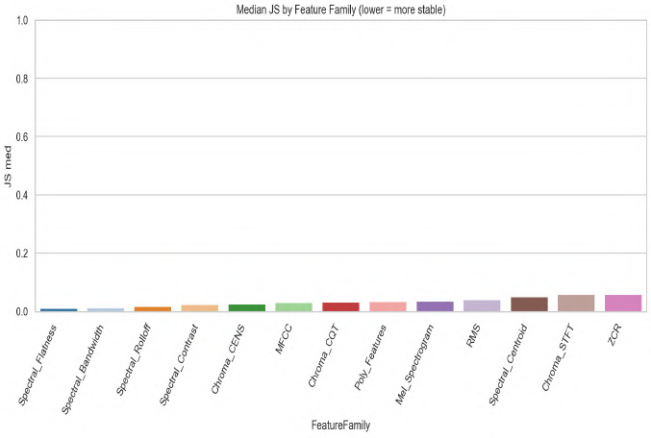


Fig. 9. Feature Family Median JS

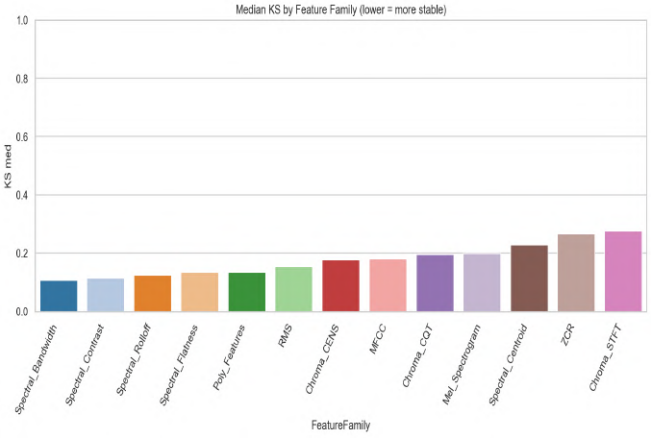


Fig. 10. Feature Family Median KS

TABLE III
MEDIAN SPEARMAN CORRELATIONS (ρ) BETWEEN DRIFT METRICS (PSI, JS, KS) AND NON-INCREMENTAL F1-WEIGHTED PERFORMANCE (QUARTERLY). NEGATIVE VALUES INDICATE THAT HIGHER DRIFT COINCIDES WITH LOWER PERFORMANCE.

Model	PSI (ρ, range)	JS (ρ, range)	KS (ρ, range)
PassiveAggressive (linear)	–0.46 (–0.53 … –0.32)	–0.48 (–0.54 … –0.33)	–0.47 (–0.53 … –0.21)
Gradient Boosting (tree)	–0.64 (–0.66 … –0.49)	–0.64 (–0.66 … –0.47)	–0.62 (–0.68 … –0.29)
Multinomial Naïve Bayes	–0.81 (–0.88 … –0.54)	–0.84 (–0.88 … –0.54)	–0.81 (–0.89 … –0.16)

classifier performance decreases as feature drift intensifies. The Passive Aggressive (linear) classifier exhibited moderate correlations ($\rho \approx$ –0.46 to –0.47), indicating measurable yet limited sensitivity to evolving distributions. The Gradient Boosting classifier demonstrates stronger effects ($\rho \approx$ –0.62 to –0.64), revealing that ensemble models degrade more sharply under drift. The Multinomial Naïve Bayes model displayed the strongest negative associations ($\rho \approx -0.81$ to -0.84), highlighting the model's high dependence on stable feature distributions and vulnerability to temporal data shifts.

Figure 11 provides a grouped bar chart showing the feature-family correlations for the Gradient Boosting model. All correlations are negative, confirming that drift consistently

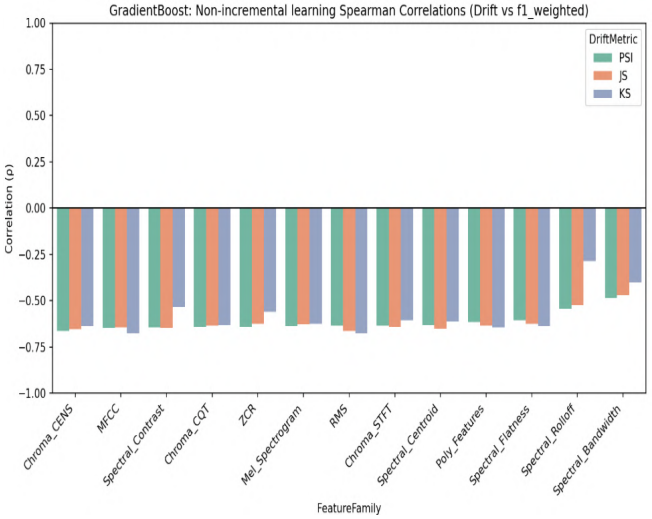


Fig. 11. Spearman Correlation (Drift vs F1_weighted)

coincides with lower F1 performance. Temporal and spectral features such as MFCC, Chroma variants, and the Mel-spectrogram displayed the strongest associations ($\rho \approx -0.63$ to -0.68), highlighting their sensitivity to evolving distributions. In contrast, the features Spectral Rolloff and Spectral Bandwidth exhibited weaker effects (around -0.29 to -0.40). Overall, the plot reinforces that features most prone to drift contribute most to model degradation under non-incremental learning.

V. DISCUSSION

The results provide evidence that audio features are not stable over time. Drift experiments showed that families such as ZCR, Chroma STFT, and Spectral Centroid are highly prone to distribution shifts, while features such as Spectral Rolloff and Spectral Bandwidth remained comparatively stable. The stacked prevalence plots revealed that some feature families drift in more than 60% of temporal transitions, and the drift versus gap curves confirmed that the drift or instability intensifies as the time interval between bins increases.

These findings have important implications for malware detection research. First, they highlight that using audio features 'as is' without any adaptation methods will result in the degradation of model performance over time. Second, not all features contribute equally to drift; some descriptors are inherently more robust, which suggests opportunities for feature selection or weighting. Finally, the negative Spearman correlations between drift scores and non-incremental model performance confirm that drift is not just a statistical artifact but has a practical impact on classifier accuracy and F1 performance.

VI. CONCLUSION

This study investigated the robustness of audio features to concept drift in Android malware detection. Using PSI, JS, and KS, the study showed that drift is pervasive and uneven across

feature families. Audio features, ZCR, and Chroma features were found to be highly unstable, while spectral features like Spectral Flatness and bandwidth were relatively stable. Negative Spearman correlations with model performance also confirm that drift degrades classifier effectiveness under non-incremental or static setups. These findings establish audio features as vulnerable to concept drift and provide a baseline for future drift-aware approaches.

VII. Future Work

While this study established that audio features suffer from concept drift in Android malware detection, future work should address mitigation. Areas for future research include incremental learning, adaptive learning (incremental or online updates), drift-triggered retraining, and feature-level selection that prioritises stable families (e.g., Spectral Flatness, Bandwidth) over drift-prone ones (e.g., ZCR, Chroma). As part of future a valuable extension would be to compare the robustness of audio features against traditional Android features such as API calls and permissions, together with exploring multimodal combinations (audio + API + images) for long-term resilience.

References

[1] Kaspersky, "Android vs. iOS Security Comparison," Jul. 2020, section: Threats. [Online]. Available: https://www.kaspersky.com/resource-center/threats/android-vs-iphone-mobile-security

[2] W. Cho, H. Lee, S. Han, Y. Hwang, and S. -j. Cho, "Sustainability of Machine Learning-based Android Malware Detection Using API calls and Permissions," in *2022 IEEE Fifth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, Sep. 2022, pp. 18–25, journal Abbreviation: 2022 IEEE Fifth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE).

[3] F. M. Darus, N. A. A. Salleh, and A. F. Mohd Ariffin, "Android Malware Detection Using Machine Learning on Image Patterns," in *2018 Cyber Resilience Conference (CRC)*, Nov. 2018, pp. 1–2. [Online]. Available: https://ieeexplore.ieee.org/document/8626828

[4] P. Tarwireyi, A. Terzoli, and M. O. Adigun, "Using multi-audio feature fusion for android malware detection," *Computers and Security*, vol. 131, no. C, Aug. 2023, number: C. [Online]. Available: https://doi.org/10.1016/j.cose.2023.103282

[5] M. Farrokhmanesh and A. Hamzeh, "A novel method for malware detection using audio signal processing techniques," in *2016 Artificial Intelligence and Robotics (IRANOPEN)*, Apr. 2016, pp. 85–91. [Online]. Available: https://ieeexplore.ieee.org/document/7529495

[6] D. Hu, Z. Ma, X. Zhang, P. Li, D. Ye, and B. Ling, "The concept drift problem in android malware detection and its solution," *Security and Communication Networks*, vol. 2017, 2017, publisher: Hindawi Limited. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85030769063&doi=10.1155%2f2017%2f4956386&partnerID=40&md5=17fc98c50a92199b95985ec14911650e

[7] Alon Gubkin, "Concept Drift in Machine Learning 101," Apr. 2021. [Online]. Available: https://coralogix.com/ai-blog/concept-drift-in-machine-learning-101/

[8] G. Hovakimyan and J. M. Bravo, "Evolving Strategies in Machine Learning: A Systematic Review of Concept Drift Detection," *Information*, vol. 15, no. 12, p. 786, Dec. 2024, publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2078-2489/15/12/786

[9] T. M. T. Pham, K. Premkumar, M. Naili, and J. Yang, "Time to Retrain? Detecting Concept Drifts in Machine Learning Systems," Aug. 2025, arXiv:2410.09190 [cs]. [Online]. Available: http://arxiv.org/abs/2410.09190

[10] F. Hinder, V. Vaquet, J. Brinkrolf, and B. Hammer, "Model-based explanations of concept drift," *Neurocomputing*, vol. 555, p. 126640, Oct. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231223007634

[11] F. Ceschin, M. Botacin, H. Gomes, F. Pinagé, L. Oliveira, and A. Grégio, "Fast & Furious: On the modelling of malware detection as an evolving data stream[Formula presented]," *Expert Systems with Applications*, vol. 212, 2023, publisher: Elsevier Ltd. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85137569528&doi=10.1016%2fj.eswa.2022.118590&partnerID=40&md5=ce4e6b4ec8ffdda606b8123971539ce5

[12] C. Gao, G. Huang, H. Li, B. Wu, Y. Wu, and W. Yuan, "A Comprehensive Study of Learning-Based Android Malware Detectors Under Challenging Environments," in *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE)*, Apr. 2024, pp. 115–127, journal Abbreviation: 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE).

[13] F. Barbero, F. Pendlebury, F. Pierazzi, and L. Cavallaro, "Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift," in *Proc. IEEE Symp. Secur. Privacy*, vol. 2022-May. Institute of Electrical and Electronics Engineers Inc., 2022, pp. 805–823, journal Abbreviation: Proc. IEEE Symp. Secur. Privacy. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85135967664&doi=10.1109%2fSP46214.2022.9833659&partnerID=40&md5=0bf2f30d2da878b21f6c98818aa25ea0

[14] A. Guerra-Manzanares, H. Bahsi, and M. Luckner, "Leveraging the first line of defense: a study on the evolution and usage of android security permissions for enhanced android malware detection," *Journal of Computer Virology and Hacking Techniques*, vol. 19, no. 1, pp. 65–96, 2023, publisher: Springer-Verlag Italia s.r.l. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85137067746&doi=10.1007%2fs11416-022-00432-3&partnerID=40&md5=c6a057ba6d71ce9c677291d35b5d0708

[15] P. Tarwireyi, A. Terzoli, and M. O. Adigun, "Meta-SonifiedDroid: Metaheuristics for Optimizing Sonified Android Malware Detection," *IEEE Access*, pp. 1–1, 2024, conference Name: IEEE Access. [Online]. Available: https://ieeexplore.ieee.org/document/10559579/?arnumber=10559579

[16] O. E. Kural, E. Kiliç, and C. Aksaç, "Apk2Audio4AndMal: Audio Based Malware Family Detection Framework," *IEEE Access*, vol. 11, pp. 27 527–27 535, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10073518

[17] S. Kumar, D. Mishra, B. Panda, and S. K. Shukla, "AndroOBFS: Time-tagged Obfuscated Android Malware Dataset with Family Information," in *2022 IEEE/ACM 19th International Conference on Mining Software Repositories (MSR)*, May 2022, pp. 454–458, iSSN: 2574-3864. [Online]. Available: https://ieeexplore.ieee.org/document/9796156

[18] T. Giannakopoulos and A. Pikrakis, "Chapter 4 - Audio Features," in *Introduction to Audio Analysis*, T. Giannakopoulos and A. Pikrakis, Eds. Oxford: Academic Press, Jan. 2014, pp. 59–103. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780080993881000042

[19] P. Raghuwanshi and R. Kaushik, "Insect Classification Using Mel-CSTFT: A Fusion of Mel Spectrogram and Chroma STFT Features," in *2024 First International Conference on Electronics, Communication and Signal Processing (ICECSP)*, Aug. 2024, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/10698728

[20] L. Nataraj, T. M. Mohammed, T. Nanjundaswamy, S. Chikkagoudar, S. Chandrasekaran, and B. S. Manjunath, "OMD: Orthogonal Malware Detection Using Audio, Image, and Static Features," Nov. 2021, arXiv:2111.04710 [cs]. [Online]. Available: http://arxiv.org/abs/2111.04710

[21] N. Harshit and K. Mounvik, "Improving Real-Time Concept Drift Detection using a Hybrid Transformer-Autoencoder Framework," Aug. 2025, arXiv:2508.07085 [cs]. [Online]. Available: http://arxiv.org/abs/2508.07085

[22] F. Bayram, B. S. Ahmed, and A. Kassler, "From concept drift to model degradation: An overview on performance-aware drift detectors," *Knowledge-Based Systems*, vol. 245, p. 108632, Jun. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122002854

[23] M. Ashawa and S. Morris, "Modeling Correlation between Android Permissions Based on Threat and Protection Level Using Exploratory Factor Plane Analysis," *Journal of Cybersecurity and Privacy*, vol. 1, no. 4, pp. 704–742, Dec. 2021, publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2624-800X/1/4/35

A Joint Federative Learning and Zero Trust-Based Privacy Enhancement Algorithm

[1]Beverly Pule, [2]Bakhe Nleyan, [3]Khulekani Sibiya,
Durban University of Technology
Faculty of Engineering and Built Environment
Department of Electronic and Computer Engineering:
[1]25120014@dut4life.ac.za, [2]bakhen@dut.ac.za, [3] khulekanis6@dut.ac.za

***Abstract*—The escalating complexity of enterprise networks—driven by diverse device ecosystems, exponential data growth, and sophisticated cyber threats—has rendered conventional perimeter-based security models obsolete. This paper proposes an integrative framework that combines Federated Learning (FL) and Zero Trust Architecture (ZTA) to address these challenges, with a focus on mitigating privacy-compromising inference attacks. FL's decentralized training paradigm aligns with ZTA's "never trust, always verify" principle, eliminating the need for centralized data aggregation and reducing the attack surface. We focus on two major threats to model privacy in such settings: property inference attacks (PIAs) and membership inference attacks (MIAs). To counter these, we introduce a Federated Learning–based Privacy Enhancement (FL-PE) algorithm. Our evaluation demonstrates that the proposed framework strengthens privacy against inference attacks while preserving high model accuracy and incurring only low-to-moderate computational overhead.**

Keywords—Privacy, Privacy, Cybersecurity, Zero Trust Architecture, Federated Learning, Membership Inference Attacks (MIAs), Property Inference Attacks (PIAs), Model Inversion

I. INTRODUCTION

The rapid evolution of enterprise networking, fueled by modern technological advancements, has produced increasingly complex infrastructures. Most enterprises now operate hybrid architectures that integrate cloud services with on-premises resources. The widespread adoption of remote work has further blurred the boundaries between internal and external networks, reducing the effectiveness of traditional perimeter-based defenses. This shift has accelerated the adoption of Zero Trust Architecture (ZTA), a model based on the principle of "never trust, always verify." Under ZTA, no entity—whether user, device, or application—receives implicit trust, regardless of its network location. Instead, access decisions are made dynamically by considering diverse inputs such as access control policies, identity repositories, user behavior logs, and threat intelligence. To remain effective, ZTA must continuously adapt to changing conditions, [1]. While frequent policy updates are essential, relying solely on human administrators is both resource-intensive and prone to error. Artificial intelligence (AI) provides an effective mechanism to automate and optimize policy enforcement. However, because ZTA environments often restrict sensitive data sharing, conventional centralized AI approaches are unsuitable. Federated Learning (FL) addresses this challenge by enabling collaborative model training without requiring direct data exchange, thereby preserving privacy while enhancing security.

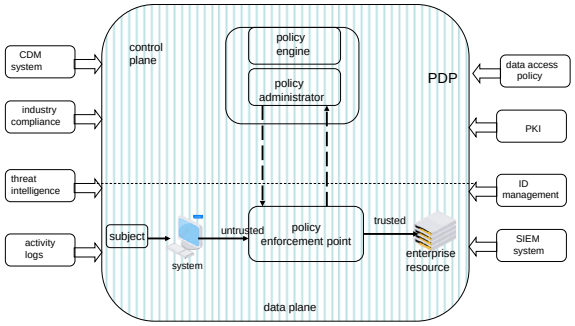


Fig.1. ZTA Core Logical Components

As illustrated in Fig. 1, ZTA deployments consist of multiple logical components that can be hosted on-premises or in the cloud. The central element, the Policy Decision Point (PDP), is subdivided into the Policy Engine (PE) and Policy Administrator (PA), which communicate across a control plane, while user data flows through a separate data plane. The purpose of ZTA is to strengthen access control and minimize risks to organizational resources through continuous monitoring and adaptive defense strategies. According to the National Institute of Standards and Technology (NIST) [1], ZTA requires applying dynamic policies as close as possible to critical resources, reducing unnecessary exposure of sensitive information such as credentials.

At the core of ZTA, policies are enforced using inputs such as user attributes, role assignments, historical activity, and external threat intelligence. These factors are managed within a policy database that supports dynamic decision-making and ongoing monitoring. Yet, strict data-sharing restrictions often lead to fragmented data silos, which hinder AI adoption. FL overcomes this limitation by allowing multiple entities to contribute to model training without exposing raw data—only model parameters are exchanged.

Building on these foundations, this paper proposes the Federated Learning–based Privacy Enhancement (FL-PE) algorithm. The main contributions of this work include the following: Demonstrating the feasibility of applying FL-based AI techniques to ZTA, Justification of FL's potential to provide an effective balance between privacy and security in ZTA environments, presentation of an architectural framework that enables secure collaboration while maximizing stakeholder benefits, and finally, proposing and evaluating the FL-PE algorithm.

II. ZTA, FL, AND DISTRIBUTED AI OVERVIEW

A. *ZTA Principles and Components*

ZT is a globally recognised security model that eliminates implicit trust and instead verifies identity and context for every access request, whether originating inside or outside the enterprise network. Unlike perimeter-based defenses that rely on location or asset ownership, ZTA enforces security on a per-transaction basis, ensuring that no subject is trusted by default. According to NIST, ZTA principles serve as the foundation for designing modern enterprise network infrastructures. The NIST reference model identifies three key logical components, as illustrated in Fig. 1, which is provided in the introductory section [1], [2].

A *Policy Engine (PE)* that hosts the zero-trust algorithm responsible for access decisions (grant, deny, or restrict). It leverages enterprise-defined policies alongside external inputs such as threat intelligence feeds, activity logs, and Continuous Diagnostics and Mitigation (CDM) systems. The PE communicates its decisions to the PA. This is followed by a *Policy Administrator (PA)* that **s**erves as the intermediary between the PE and enforcement points. It translates PE decisions into actionable configurations, may issue tokens or session credentials, and configures the Policy Enforcement Point (PEP).

Next is a *Policy Enforcement Point (PEP*), which mainly **a**cts as the gateway between untrusted users or devices and enterprise resources. It establishes, monitors, and terminates connections, enforcing dynamic access decisions in real time.

At the heart of ZTA is the ZT Algorithm within the PE that validates each access request individually, even after authentication and authorization.

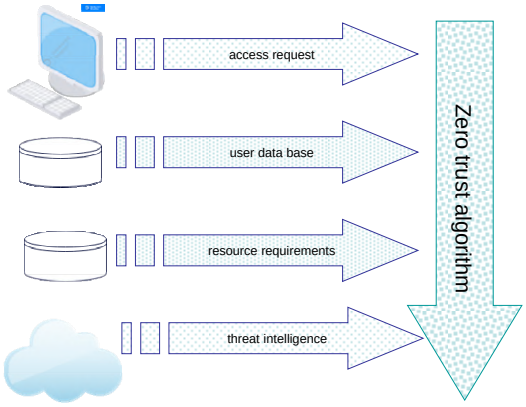


Fig. 2. Zero Trust algorithm.

Unlike static, rule-based approaches, the algorithm must be continuously updated to reflect evolving threats and contexts. NIST broadly classifies the required data sources into four categories, as illustrated in Fig. 2.

- *Subject Database***:** Credentials, roles, attributes, and privileges.
- *Asset Database***:** Enterprise and non-enterprise assets, including device health, software versions, integrity status, and location.
- *Resource Requirements***:** Enterprise policies, including authentication levels, data sensitivity thresholds, and MFA requirements.
- **T**hreat Intelligence Feeds**:** Real-time data on malware activity, suspicious queries, and known attack signatures.

Manual maintenance of the trust algorithm is impractical given the scale of data and inputs. AI can automate adaptive trust algorithms, but centralized approaches introduce privacy and regulatory challenges. This motivates Distributed Artificial Intelligence (DAI) as a privacy-preserving alternative.

AI-based ZTA Implementations, wise, AI-enabled ZTA approaches exist, integrating context-aware learning with enforcement points. However, most rely on centralized data, which limits both privacy and scalability. Federated Learning addresses this gap by enabling decentralized, privacy-preserving training. Given the scale and diversity of these inputs, maintaining and updating the trust algorithm manually is impractical. AI has therefore been proposed as a mechanism to automate the development of adaptive trust algorithms. However, issues such as privacy concerns, distributed databases, and the cost of centralizing sensitive data continue to be major obstacles. In large, geographically distributed enterprises—or in multi-party collaborations—aggregating data into a single repository is often infeasible or undesirable due to privacy, cost, or regulatory restrictions. This motivates the use of Distributed Artificial Intelligence (DAI) techniques to design intelligent trust algorithms without requiring centralized data sharing.

Several AI-enabled ZTA approaches have been explored in the literature, e.g the work in [x] proposed and *intelligent ZTA (i-ZTA)* for next-generation communication systems. Their architecture integrated AI with the PEP and Policy Decision Point (PDP) to support context-aware decision-making using reinforcement learning. While effective in dynamic access control, their framework relied on centralized data aggregation, which raises privacy concerns. The authors in [3] developed an AI-driven ZTA for smart healthcare environments, focusing on policy enforcement and trust assessment. Similar to [4], their design required centralized access to sensitive databases to train the trust models. The authors in [5] applied AI classification models at the PEP to accept or deny access requests. Although their method demonstrated feasibility, it lacked dynamic contextual updates and was therefore unsuitable for rapidly changing environments. Furthermore, their reliance on centralized data storage made the solution vulnerable to security and privacy risks.

In summary, while AI enhances the adaptability and efficiency of ZTA, most existing approaches depend on centralized access to sensitive data, making them impractical for distributed and privacy-sensitive environments. This limitation motivates the exploration of federated and distributed learning techniques as a privacy-preserving alternative.

B. *Distributed Artificial Intelligence*

DAI refers to AI systems where both model training and inference are performed in a decentralized manner, typically close to the data sources. Introduced in the 1980s, DAI has grown substantially with advances in parallel computing and

distributed architectures. Its main advantage is the ability to leverage parallelism to accelerate learning and efficiently utilize computing resources. Parallelism in DAI is generally categorized into two main types [4], [5];

Data Parallelism: This method divides a large dataset into smaller segments, distributing each segment across multiple nodes for parallel computation. It is especially useful when individual nodes cannot process the entire dataset independently, such as in Internet of Things (IoT) networks. However, in enterprise environments with sufficient local computing resources, it is often preferable to train models near the data itself. For Zero Trust Architecture (ZTA), this approach ensures sensitive data remains within its local environment, supporting privacy and security requirements.

Model Parallelism: Instead of splitting the dataset, model parallelism partitions the AI model itself into smaller components, which are trained across multiple nodes. For example, different layers of a neural network may reside on different nodes and be trained collaboratively. Model parallelism is particularly advantageous for ZTA because it allows model learning without moving sensitive data, thus preserving privacy. Federated Learning (FL) is a practical implementation of this approach, where local models are trained on separate nodes and only parameter updates are shared with a central aggregator, avoiding direct exposure of raw data.

Overall, while both data and model parallelism facilitate distributed learning, model parallelism—especially in the form of FL—offers stronger privacy guarantees. This makes it particularly suitable for applications like ZTA, where data confidentiality is critical.

C. *Federated Learning*

FL is a distributed AI approach that enables collaborative model training across multiple data sources without sharing raw data, [6], [7]

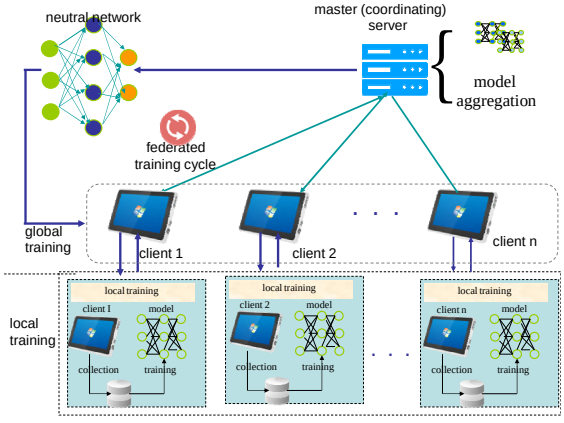


Fig. 3. Federated Learning architecture.

Instead of centralizing sensitive information, FL trains models locally and exchanges only model updates or parameters, which are aggregated to improve a global model. This is illustrated by Fig. 3.

Its key characteristics include;

*Privacy and Data Protection***:** FL allows collaborative learning from scattered datasets while keeping sensitive information local, aligning with NIST's guidance on data minimization and privacy protection in ZTA.

*Open-Source Frameworks***:** Several frameworks, such as FATE, PySyft, PaddleFL, and FedML, support large-scale deployment of FL and can be used to develop trust algorithms In terms of classification**,** FL can be organized into two architectural categories:

centralized Federated Learning (CFL): Involves a central server coordinating the aggregation of client updates and redistributing the refined global model (Fig. 4). Its main limitation is the single point of failure in the central server.

Decentralized (Distributed) Federated Learning (DFL): Removes the central aggregator. Clients operate in a peer-to-peer network, exchanging model parameters directly. Each participant refines its local model independently, without a single global model.

FL's distributed, privacy-preserving nature makes it a strong candidate for implementing AI-driven ZTA algorithms, enabling adaptive, secure, and collaborative access control across geographically and organizationally distributed networks.

In summary, together, ZTA, DAI, and FL form a complementary foundation for modern enterprise security:

- ZTA provides the security framework with continuous verification and dynamic access control.
- DAI enables intelligent decision-making in distributed environments while reducing reliance on centralized data.
- FL supports collaborative model training that preserves privacy and aligns with the principles of ZTA.

Integrating these technologies allows enterprises to implement **adaptive, scalable, and privacy-conscious zero-trust algorithms** capable of responding to complex, hybrid, and distributed network environments.

III. MEMBERSHIP INFERENCE-RELATED ATTACKS IN FL

As FL is still in its infancy, it thus possesses several significant challenges, particularly in terms of maintaining data privacy and utilizing a decentralized approach for both local and global learning processes. As is known, its primary objective is to collaboratively construct a global learning model without requiring the sharing of actual data from various distributed sources. This method usually involves a coordinating entity that manages the necessary information exchange for training the unified model. Participants can then modify the model parameters based on their data and share those updates with others, promoting a collaborative, localized, and decentralized machine learning environment.

A pertinent privacy-related challenge is implementing strategies that combat interference. These are broadly categorized as those that target the model updates shared during training (input privacy) versus those that extract data from the AI model after training is completed (output privacy). Input privacy strategies prevent the adversary from learning anything about the *inputs* (i.e., the model updates) to the system and are mostly cryptographic-based. Output privacy strategies provide post-AI-trained model protection from the same

adversary attacks and often consist of *model parameters* that control the model's predictions.

As illustrated by Fig. 3, the training data sets held by clients 1 and 2 are respectively;

$$Y_1 = \left\{ y_1^1, y_2^1, ... y_n^1 \right\} \quad (1)$$

and

$$Y_2 = \left\{ y_2^2, y_2^2, ... y_n^2 \right\}, \quad (2)$$

Generally ;

$$Y_1 \cap Y_2 = \varnothing \quad (3)$$

Single clients' data is often insufficient to support the training of a practical machine learning model in a federated learning system. For privacy reasons, clients also do not want to hand over their local data directly to other clients or the cloud server for training models. Therefore, after reaching a consensus on the model structure with other clients, each client only needs to utilize their local data to train a local model. Then upload the gradients or parameters of the local model to the server for aggregation to obtain a practical global model.

$$\nabla \Omega_{global} = n^{-1} \sum_{i=1}^{n} \nabla \Omega_i \quad (4)$$

After the aggregation is finished, the cloud server delivers the aggregated model (global model) to each client for retraining and repeats this process until the training converges

As such, inference attacks can be divided into membership inference attacks (MIAs), property (data distribution) inference attacks (PIAs), model inversion attacks (MIs), and model inference attacks[8], [9], [10].

(*MIAs)* attempt to determine whether a record exists in the target model's training set. They can be classified into either binary or metric-based attacks. Essentially, binary classification-based MIAs are trying to utilize a machine learning model (attack model) to find the difference in the behavior of the target model on members and non-members. Specifically, the main idea is that the attacker creates and trains multiple "shadow models" to simulate the behavior of the target model (i.e., if the target model is a facial classifier, then these "shadow models" are also facial classifiers), and then labels the outputs of these "shadow models" (e.g., label 1 means that the input corresponding to this output is from the training set, and vice versa). Finally, based on the labeled output, attackers train a binary classifier that can identify whether the input corresponding to the model output is a member. However, training such a binary classifier is a challenging task.

On the other hand, metric-based models first perform a specific metric operation on the target model's outputs and then compare the metric value with a predefined threshold to determine whether the target data corresponding to the outputs belongs to the target training set [11]. They are categorized into prediction correctness, prediction loss, prediction confidence, adversarial perturbation, and prediction entropy. The prediction entropy is defined by the following equation:[13].

$$P[\mathbf{E}](\hat{p}(j|i), j) = -(1 - p_j)\log(p_j) - \sum_{z \neq i} \log(1 - p_z) \quad (5)$$

Where $\hat{p}(j|i)$ is the confidence corresponding to data j output target model, and p_z are elements in $\hat{p}(j|i)$.

PIAs mainly use publicly available information about a specific instance to infer unknown details of that instance, such as sexual orientation, political tendency, gender, and so on, within online social networks. Later, the concept of PIA was expanded to infer the overall properties of the training set, like gender ratio and age distribution, and it gradually became mainstream. Specifically, they assume the attacker has an auxiliary dataset and split it into two parts based on whether the data possesses a certain property or not. During each model update, attackers use the model snapshot to get the gradients associated with each part and then label them accordingly. The attack model is finally trained on this labeled dataset. In the attacks described above, the attacker does not interfere with the training of the target model, which is called a passive attack. When attackers interfere with the training process, often to boost the effectiveness of the PIA, these are known as active attacks. In terms of classification, we distinguish the following: *active models*: This underlies an intuition that the data generated by a generative type model can reflect certain properties of the training set. Therefore, the attacker can process the data generated by the target generative model to obtain the underlying properties of the target dataset. Its generalized three-step strategy is summarized as follows:

- sampling the latent codes $Y = y_1 y_2, ... y_n$ and ultimately utilizing it to query a target adversarial generative network to gain access to generated samples $X = x_1, x_2, ... x_n$.
- Adversary constructs a property classifier X
- The adversary develops a function ξ that computes and characterizes a certain property of target data distribution based on the property classifier's outputs, defined as;

$$\xi(\{Q_{property}, < classifier. >, (\chi_{t\arg et}(y))\} \Big|_{i=1}^{Y} \quad (6)$$

Ultimately, they utilize shadow training techniques to refine the latent code, making the attack more effective.

Embedding Models: These models learn from distributed data while preserving privacy; their goal is to infer properties of a specific target model's input rather than statistics of the target training set.

Attribute Inference Attacks (AIAs): these are in three categories as follows 12], [13]:

Friend-based attribute inference, which mainly take advantage of the homogeneity between data. The attack mainly consists of two steps: 1) The attacker first collects the friend list of the target user; 2) Then, the attacker infers the attributes of the target through the public information of the target user's friends.

Behavior-based attribute inference. The core idea of behavior-based attribute inference is: you are how you behave. For example, an attacker can judge a target user's private attributes through him/her behavior, such as liking, following, and joining a group chat. Specifically, they construct the user's rating behavior as a feature vector. The value of the ith item of this vector is equal to the user's rating for the ith movie,

and if the user has not seen this movie, the corresponding value is 0.

Hybrid attribute inference. The two types of methods mentioned above are based on either social structure or behavior, but not both. Thus, this category combines both, and thus in the process can yield better accuracy in its intended objectives. i.e., of performing accurate attribute inference.

Reconstruction Attacks (RAs): these focus on ways to recover unstructured data such as images, audio, and video.

MIs focus on reconstructing data in the training set of machine learning models, such as a training image, a sentence, etc. They can be categorized into attribute inference attacks (AIAs) versus reconstruction attacks (RAs). Both models aim to infer specific data information, such as a person's age and image content. Mostly they rely on gradient fitting methods. i.e. in a typical Gradients/Parameters-based attack, the adversaries will design a fitting scheme to fit the obtained gradients, and use it as a loss function to iteratively optimize a certain noise to

reconstruct the victim's privacy. These gradient-fitting-based methods in FL are usually based on some strong assumptions. For example, the attacker needs to know the target data's batch size or the activation function type, which makes these attacks impractical.

Model inference attacks (also known as model extraction model, (MEAs) aim to obtain a duplicate of the cloud model through outputs of the cloud model. Since building a cloud model is a costly business behavior, and the cloud model is a crucial and sensitive business asset, the threat of model inference attacks cannot be overlooked. In other words, MEAs aim to steal the privacy of trained models, such as parameters, structure, capabilities, etc. Three approaches to extracting the target model include;

- *Extract model parameters*. "Parameters" represent the "weight parameters" of each node of the model and the "hyperparameters (such as learning rate)" during model training. This attack method usually assumes that the attacker knows the structure of the model (or at least the model type).
- *Extract model structure*: After knowing the model structure, the attacker can carry out more threatening attacks.
- *Extract model functionality*. Unlike the previous two methods, this approach focuses on imitating the model's behavior, such as classification, generation, and regression, rather than its attributes.

Overall, based on our observations, "inference attacks" are evolving towards stealthiness, making them harder to detect. In addition, the defender is relatively passive in this privacy game; that is, the attack method appears before the counter strategy

IV. FEDERATED LEARNING BASED PRIVACY ENHANCEMENT ALGORITHM

The decentralized, collaborative nature of Federated Learning (FL) enables participating clients to train models locally using private data, while only sharing model parameters with a centralized server for aggregation and consolidation into a global model. This approach inherently enhances privacy. However, FL-based training systems remain vulnerable to Membership Inference Attacks (MIAs), which attempt to extract sensitive information about the training data from the exchanged models.

To mitigate this vulnerability, various cryptographic techniques—often combined with secure computation protocols and deployed within otherwise Zero-Trust (ZT)–based trusted environments—have been explored.

In this section, we propose and evaluate a**,** Federated Learning-based Privacy Enhancement (FL-PE) algorithm designed to resist membership inference attacks while optimally balancing privacy, utility, and computational cost. Our approach takes into account that membership privacy leakage primarily occurs at the FL model layer.

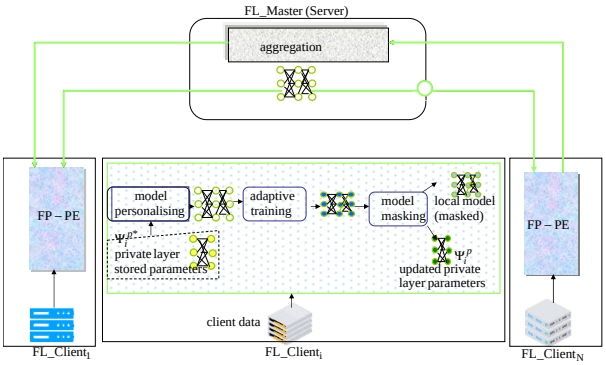


Fig. 4. FL-PE algorithm structure

The implementation takes place on the client side to protect both the overall federated learning (FL) model and individual client models. It introduces a privacy-sensitive layer that masks and safeguards model updates before they're sent to the FL server. Consequently, the server only receives an aggregated global model containing the masked version of this layer. When clients receive the global model, they restore their original, unmasked privacy-sensitive layer—preserved from the previous FL round—and integrate it with the global model to create a personalized model for predictions. The goal is to assess how different model layers influence Membership Inference Attacks (MIAs), which aim to identify whether a data sample was included in the training data. Using a trained FL model, we examine predictions on both member and non-member samples, calculate gradients for each layer, and analyze the generalization gap—the difference between member and non-member gradients. A larger gap suggests a stronger MIA signal, making membership identification easier.

Fig.4 illustrates the overall algorithm structure, where each client executes the process independently. The FL-PE privacy-preserving strategy targets the penultimate layer, called the privacy-sensitive layer, as it is most susceptible to leaking personal information. While this layer is used locally for predictions, it is always encrypted before sending updates to the FL server. As a result, the global model stored by the server never contains the raw privacy-sensitive layer. When clients receive the updated global model, they reinsert their previously stored non-obfuscated version, creating a personalized model that balances privacy and predictive performance.

Algorithm: FL-PE algorithm on client's side

Initialisa tion : $\sum \psi$: set of curent global model parameters
Initialisa tion : $\sum \psi_i$: set of curent client model parameters
Local variables
ψ_i^{p*} : client layer's private layer parameters
$(\Gamma^i, \mathrm{Y}) = \{(\Gamma_1^i, \mathrm{Y}_1)\}$: client$_i$'s training phase
I : learning rate
# Model personaliz ation
{
for $\overline{1,...j}$ **do**
*i***f** $j \neq p$ **then**
// call global model's j^{th} layer parameters;
$\Psi_i^j \leftarrow \Psi^j$
else
// client' s private layer parameters are restored
$\Psi_i^j \leftarrow \Psi_i^{p*}$
// Adaptive Model Training
// setting of initially accumulate d grdients matric
$\Omega \leftarrow 0$
*f***or each** local training cycle **do**
*f***or each** $(\Gamma_k^i, \mathrm{Y}_k) \in (\Gamma^i, \mathrm{Y}))$ **do**
// Peforming Local Predction
$\hat{\mathrm{Y}}_k \leftarrow \Psi_i(\Gamma_k^i)$
// computing new accumulate d gradients
$\Omega \leftarrow \Omega + \nabla \xi_\Psi(\mathrm{Y}_k, \hat{\mathrm{Y}}_k)$
// updating local model & adaptive grdient degrdation
$\Psi_i \leftarrow \Psi_i - \mathrm{I} \frac{\nabla_\Psi .loss}{\sqrt{\Omega} + 10^{-5}}$
// Model masking
// Client' s private layer parameters saved
$\Psi_i^j \leftarrow \Psi_i^{p*}$
// Masking Client' s private layer parameters
$\Psi_i^{p*} \leftarrow$ randomised values
return Ψ_i
}

To evaluate the proposed algorithm, we utilized a range of datasets encompassing image recognition, speech processing, and structured tabular tasks. This variety ensures the evaluation reflects both domain-specific performance and the overall generalizability of the approach [14], [15],[16].

Image datasets: The CelebA dataset, consisting of approximately 203,000 facial images, was used for recognition and attribute detection tasks. For experimentation, we created a resized subset of 500,000 samples at 32×32 pixels and derived 32 binary attribute classes that reflect characteristics such as gender, skin tone, spectacles, and facial features (including lip and nose size). Additionally, we prepared two CIFAR-derived datasets, each comprising 50,000 images at 16×16 resolution. The first subset, CIFAR-25, includes 25 classes, while CIFAR-200 covers 150 categories across animals, vehicles, and everyday objects. To account for real-world visual variability, we incorporated the German Traffic Sign Recognition Benchmark (GTSRB), which contains 52,000 images across 45 traffic sign categories. This dataset captures illumination, weather, and camera-angle variations, making it particularly suitable for evaluating models in safety-critical autonomous driving contexts.

Audio datasets: For sequential and acoustic data, we utilized the Speech Commands dataset, which includes ~65,000 short utterances recorded by more than 2,000 speakers. These recordings span 40 word classes. The classification task was performed using the M18 convolutional neural network (24 layers, 4.1M parameters), a widely adopted model for keyword spotting.

Tabular datasets: To evaluate structured data, we used Purchase100, which contains 100,000 records with 600 binary attributes representing consumer purchase behaviour. The task involves classifying customers into 100 distinct categories. Similarly, the Texas100 dataset, derived from inpatient discharge records of the Texas Department of State Health Services, was employed. It includes details such as diagnoses, causes of injury, surgical interventions, and demographic attributes (age, gender, race). The same classification architecture applied to Purchase100 was also used for Texas100.

V. Performance Results and Discussion

Both analytical and simulation evaluation approaches are carried out. In our both cases we benchmark our proposed framework against privacy-preserving mechanisms investigated in [17], namely Local Differential Privacy (LDP), Central Differential Privacy (CDP), and Weak Differential Privacy (WDP). These mechanisms were originally formulated to attenuate vulnerabilities in federated learning (FL) pipelines, particularly susceptibility to backdoor insertion and inference-based adversarial analysis. Prior studies have established that perturbation-oriented schemes constitute an effective defense paradigm, yielding statistically significant reductions in attack success rates while preserving model utility under varying threat models.
To assess the performance of the algorithm, we utilize several of the following performance metrics.

Attack Success Rate (ASR): Quantifies the proportion of adversarial attacks that succeed in misleading the model, thereby assessing its robustness.

Attack AUC (Area Under the Curve): Evaluates resilience to membership inference attacks (MIAs). A score of 50% reflects random guessing, while 100% indicates perfect inference. Effective defenses aim to approach 50%.

Over*all Model Privacy Metric (OMPM):* Captures the highest potential privacy leakage from both global and client-side models.

Overall Model Utility Metric: Defined as predictive accuracy, i.e., the proportion of correctly classified samples relative to the total dataset size.

We begin by examining how FL-PE defends against membership inference attacks (MIAs) on both local and global models. Each dataset is split into training and test sets, with half of the data allocated as the adversary's prior knowledge

and the remaining portion used for evaluation. We benchmark FL-PE against common defenses, namely WDP, LDP, and CDP.

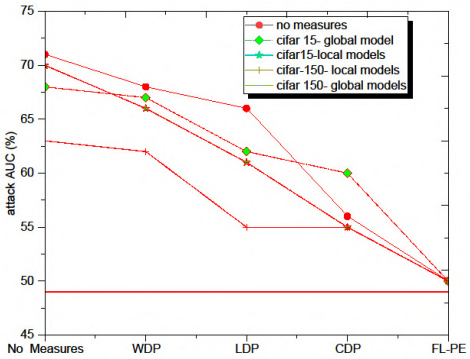


Fig. 5.Comparative Analysis of Privacy Leakage: FL-PE vs. State-of-the-Art Protection Mechanisms

Figures 5–7 present the attack AUC for local and global models separately, with each bar representing a different defense. Across all evaluated datasets, FL-PE consistently lowers the attack AUC toward the random baseline of 50%, showing a strong level of privacy protection.

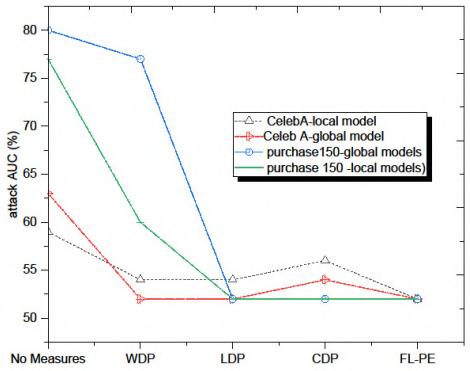


Fig. 6. Comparative Analysis of Privacy Leakage: FL-PE vs. State-of-the-Art Protection Mechanisms cont'

Differential privacy methods, by contrast, demonstrate less reliable performance. For instance, on local models, FL-PE achieves up to a 29% reduction in privacy leakage (Figure 5), whereas WDP yields only 3% and CDP reaches 25%.
This effectiveness arises from FL-PE's design: sensitive layers are concealed, and parameters are substituted with randomized values. The resulting perturbation ensures that adversaries observe a distorted version of the target model, making its outputs difficult to align with those of the shadow model. This undermines the attack process and explains the observed drop in AUC. To assess the influence of FL-PE on model behavior, we study the loss profiles of member and non-member samples under each defense. Ideally, these distributions should overlap—preventing membership signals—while remaining concentrated around low loss values to reflect good predictive performance. On the Purchase100 dataset, we compute the loss separately for members and non-members across all defense settings.

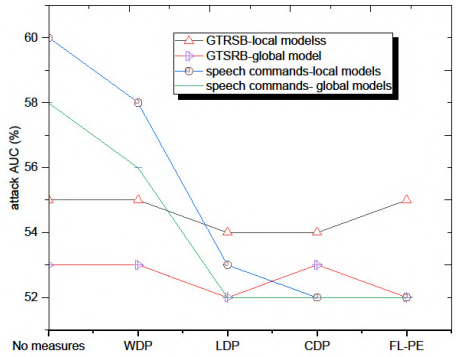


Fig. 7. Comparative Analysis of Privacy Leakage: FL-PE vs. State-of-the-Art Protection Mechanisms cont'

The results show that FL-PE effectively reduces divergence between the two distributions while retaining mostly low loss values. Differential privacy mechanisms, however, either fail to close the gap or cause substantial accuracy loss.

Finally, for computational efficiency, we analyze the resource costs of FL-PE relative to differential privacy. We consider three metrics: client training time per round, server aggregation time, and GPU memory usage.

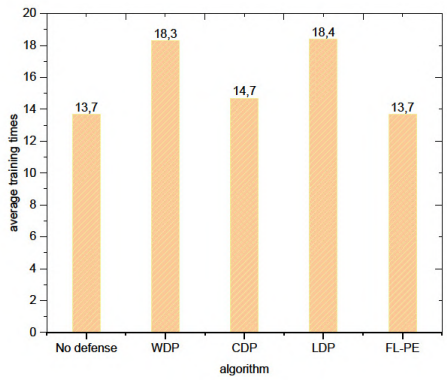


Fig. 8. Client-side model training time frames.

As illustrated in Fig. LDP, WDP and CDP introduce significant slowdowns due to noise addition and gradient perturbation. FL-PE, however, incurs negligible computational cost, since its protection mechanism relies on lightweight parameter obfuscation rather than expensive gradient manipulation. This enables FL-PE to maintain efficiency while ensuring robust privacy, effectively addressing one of the primary drawbacks of differential privacy in FL We further evaluate the temporal cost of server-side aggregation across different privacy-preserving configurations. Aggregation latency is defined as the elapsed time between the reception of all client

model updates and the transmission of the aggregated model. The simulation results presented in Fig. 9 demonstrate that Central Differential Privacy (CDP) incurs a substantial latency overhead, primarily due to its inherent mechanism of injecting calibrated noise into the global parameter set prior to dissemination. This operation significantly prolongs the aggregation process, with delays observable at the scale of seconds in our experiments.

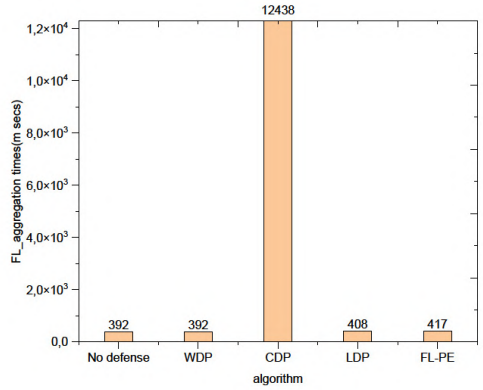


Fig.9. Server-side FL aggregation time frames

Conversely, Federated Learning with Parameter Encryption (FL-PE), Local Differential Privacy (LDP), and Weak Differential Privacy (WDP) exhibit aggregation times of comparable order to the non-privacy-preserving baseline, indicating that these approaches introduce negligible additional computational burden during aggregation.

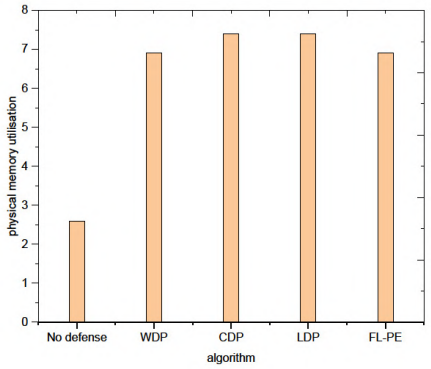


Fig.10. Client-side memory utilisation

We further evaluate GPU memory utilization, with an emphasis on the overhead induced by differential privacy (DP) mechanisms. Local DP, Central DP, and Weak DP each impose substantial memory requirements, leading to a 159% increase compared to a no-defense baseline. This overhead stems primarily from (i) storing calibrated noise vectors, (ii) maintaining privacy budget bookkeeping, and (iii) buffering model updates prior to perturbation. As can be observed from Fig. 9, in contrast, FL-PE avoids noise injection and budget management by design, and thus incurs negligible GPU memory overhead, making it memory-optimal relative to DP-based defenses.

VI. CONCLUSION

In this paper, we introduced and evaluated **FL-PE**, a novel defence framework designed to tackle one of the most pressing privacy risks in federated learning: membership inference attacks. Unlike existing solutions, FL-PE was tested across a wide range of datasets, neural network architectures, and advanced privacy-preserving mechanisms, showing consistent performance. Our results demonstrate that FL-PE not only enhances privacy but also preserves model accuracy while maintaining low computational and communication costs—making it both effective and practical for real-world deployment.

Importantly, the benefits of FL-PE extend beyond membership inference. The framework shows strong potential to defend against broader privacy threats, such as property inference and model inversion attacks, positioning it as a versatile tool for safeguarding federated learning systems. Looking ahead, we see exciting opportunities in developing automated methods to pinpoint the neural network layers most vulnerable to privacy leakage, tailored to specific attack types, threat models, and FL architectures. Such advances could pave the way toward more adaptive, resilient, and intelligent privacy defences in federated learning

REFERENCES

[1] NIST, “Zero Trust Architecture,” Special Publication 800-207, Aug. 2020.

[2] A. Soni, S. K. Nanda, R. Priyadarshini and P. K. Meher, "A Platform-agnostic Solution for a Pluggable and Customizable Approach to Zero Trust Architecture," 2025 International Conference on Advanced Computing Technologies (ICoACT), Sivalasi, India, 2025, pp. 1-7, doi: 10.1109/ICoACT63339.2025.11005274

[3] A. Alshehri and C. Tunc, "Zero Trust Engine for IoT Environments," 2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), Giza, Egypt, 2023, pp. 1-3, doi: 10.1109/AICCSA59173.2023.10479286.

[4] B. L. Vennala, D. Suresh, P. Samiya, M. Suresh and B. Vamsi, "Integrating Artificial Intelligence with Cloud Platforms to Optimize Performance, Scalability, and Reliability in Distributed Computing Systems," 2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI), Chennai, India, 2025, pp. 1-6, doi: 10.1109/ICDSAAI65575.2025.11011769.

[5] H. Xu, K. P. Seng, L. M. Ang and J. Smith, "Decentralized and Distributed Learning for AIoT: A Comprehensive Review, Emerging Challenges, and Opportunities," in IEEE Access, vol. 12, pp. 101016-101052, 2024, doi: 10.1109/ACCESS.2024.3422211.

[6] A. Saha, L. Ali, R. Rahman, M. F. Monir and T. Ahmed, "Exploring Federated Learning: The Framework, Applications, Security & Privacy," 2024 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Tbilisi, Georgia, 2024, pp. 272-275, doi: 10.1109/BlackSeaCom61746.2024.10646291.

[7] A. Rahdari et al., "A Survey on Privacy and Security in Distributed Cloud Computing: Exploring Federated Learning and Beyond," in IEEE Open Journal of the Communications Society, vol. 6, pp. 3710-3744, 2025, doi: 10.1109/OJCOMS.2025.3560034.

[8] P. Yu, K. Song and J. Lu, "Generating Adversarial Examples With Conditional Generative Adversarial Net," 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 2018, pp. 676-681, doi: 10.1109/ICPR.2018.8545152.

[9] M. Mirza and S. Osindero. “Conditional Generative Adversarial Nets”. *ArXiv.* , 2014https://arxiv.org/abs/1411.1784, 2014

[10] Fan Mo et al. “Layer-wise Characterization of Latent Information Leakage in Federated Learning”. The International Conference on Learning Representations(ICLR) - Workshop on Distributed and Private Machine Learning. Virtual,2021.

[11] F. Mo, A.Borovykh, A, M. Malekzadeh, H. Haddadi and S.Demetriou. “ Layer-wise Characterization of Latent Information Leakage in Federated Learning”. 2020, *ArXiv*. https://arxiv.org/abs/2010.08762

[12] F. Mo, A.Borovykh, A, M. Malekzadeh, H. Haddadi and S.Demetriou. (2021). “Quantifying Information Leakage from Gradients”. *ArXiv*. 2021, ttps://arxiv.org/abs/2105.13929.

[13] C. Zitao and K. Pattabiraman. “ Overconfidence is a Dangerous Thing: Mitigating Membership Inference Attacks by Enforcing Less Confident Prediction”, Network and Distributed System Security (NDSS) Symposium 2024 26 February- 1 March 2024, San Diego, CA, USA ISBN 1-891562-93-2
https://dx.doi.org/10.14722/ndss.2024.23014.

[14] P. Warden, P. “Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition”. *ArXiv*. https://arxiv.org/abs/1804.03209.

[15] https://www.kaggle.com/datasets/jessicali9530/celeba-dataset

[16] https://www.cs.toronto.edu/~kriz/cifar.html (CIFAR 10 and CIFAR 100 data sets)

[17] M .Naseri,J. Hayes, and E. De Cristofaro, “Local and Central Differential Privacy for Robustness and Privacy in Federated Learning”., Published in the Proceedings of the 29th Network and Distributed System Security Symposium (NDSS 2022)2020, *ArXiv*. https://arxiv.org/abs/2009.03561

Enhancing IoT Security with Anomaly Detection: An Interpretable Multilayer Perceptron Approach

Nathan Morgenrood
School of Computer Science and Information Systems
North-West University
Potchefstroom
nathan.morgenrood@nwu.ac.za

Tiny du Toit
School of Computer Science and Information Systems
North-West University
Potchefstroom
tiny.dutoit@nwu.ac.za

Linda Redelinghuys
School of Computer Science and Information Systems
North-West University
Potchefstroom
linda.redelinghuys@nwu.ac.za

***Abstract*— The proliferation of the Internet of Things (IoT) introduces significant security challenges, requiring anomaly detection systems that are not only accurate but also efficient and interpretable. This study proposes a framework that automates the design of an interpretable multilayer perceptron model by integrating a custom evolutionary algorithm with SHapley Additive exPlanations (SHAP) for feature selection and transparency. Applied to a subset of the CICIoT2023 dataset, the SHAP-based feature selection method achieved a 58.06% dimensionality reduction, which resulted in a 76.4% decrease in the final model's training time. The final, lightweight model achieved a competitive accuracy of 98.58%, demonstrating a negligible performance trade-off of only 0.55% in exchange for these significant gains in computational efficiency. The framework successfully balances high accuracy with efficiency and provides clear, feature-level interpretations, fostering greater trust and reliability in AI-driven security solutions for IoT networks.**

Keywords—Anomaly Detection, AutoML, Boruta-SHAP, Evolutionary Algorithm, Explainable AI, IoT, Model Interpretability, MLP, SHapley Additive exPlanations.

I. INTRODUCTION

The Internet of Things (IoT) is a network of physical objects or "things" embedded with electronics, software, sensors, and network connectivity, allowing these objects to collect and exchange data [1]. This interconnected structure has gained popularity by linking diverse devices, from household appliances to industrial equipment [2]. It significantly enhances efficiency and automation by transforming everyday objects into smart devices that operate with minimal human intervention. However, despite the widespread adoption of IoT in various sectors, including telecommunications, the security and privacy aspects of IoT devices often remain overlooked, posing risks to data confidentiality and integrity [3].

These concerns extend beyond technical shortcomings, revealing significant gaps in integrating security at various stages of IoT development [3]. Robust security is clearly needed, but it is not always feasible or within the scope of every solution. This emphasises the importance of post-development interventions. Further, literature found on IoT security often overlooks the importance of incorporating interpretability into machine learning models. Most existing approaches focus on improving model accuracy for detecting network anomalies without emphasising model transparency and comprehensibility.

The aim of this study is to enhance the accuracy and interpretability of anomaly detection models within the IoT security domain, regardless of their built-in security levels. This is achieved through a multi-stage framework leveraging Automated Machine Learning (AutoML) and SHapley Additive exPlanations (SHAP). This combination optimises model performance and computational efficiency while simultaneously enhancing interpretability. The AutoML component automates model selection and hyperparameter tuning, reducing the time and expertise required to develop high-performing anomaly detection models [4]. Concurrently, the integration of SHAP provides a transparent understanding of the multilayer perceptron's (MLP) decision-making process in identifying and mitigating security threats [5]. This blend of AI's analytical power with SHAP-based interpretability is crucial for fostering the trust and reliability needed in modern AI-driven solutions [6].

A representative subset of the comprehensive CICIoT2023 dataset will be used to validate this framework [7], ensuring a direct and equitable comparison against a benchmark study that utilised the same data sample. This evaluation aims to demonstrate the framework's potential to enhance the resilience and intelligence of IoT cybersecurity measures.

The remainder of this article is structured as follows. In Section II, the necessary background on IoT security, MLPs, and interpretable AI is reviewed. The proposed evolutionary framework for automated model design and feature selection is presented in Section III. The experimental setup, including the CICIoT2023 dataset, pre-processing, and evaluation metrics, is detailed in Section IV. In Section V, the results are presented and discussed, where the performance, efficiency, and interpretability of the baseline and proposed models are compared against benchmark methods. Finally, the article is concluded in Section VI with a summary of the findings and their significance.

II. BACKGROUND

Essential concepts relevant to the study will be introduced in this section. The focus will be on applying deep learning techniques for anomaly detection in IoT security, particularly using MLPs. The role of AutoML in simplifying complex model optimisation tasks will be discussed. Additionally, the necessity for transparent and comprehensible decision-making processes in neural networks to ensure accountability and trustworthiness will be explained.

A. Anomaly Detection in IoT Security

Anomaly detection is essential within IoT security, focusing on identifying deviations from normal operational patterns to enable early threat recognition and mitigation [8]. The significance of anomaly detection lies in its capacity to pre-emptively flag potential intrusions or attacks, ensuring the security and integrity of data before substantial harm occurs [3]. Distinct from traditional security mechanisms, anomaly detection uniquely positions itself against zero-day exploits by not depending solely on known threat signatures [9]. This adaptability is particularly valuable in the IoT domain, where the diversity and proliferation of devices continuously introduce new vulnerabilities.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

By leveraging deep learning models, like MLPs, anomaly detection systems are enhanced to accurately learn and adapt to the normal behaviours of IoT environments [8]. These models are adept at analysing complex data from myriad IoT devices, facilitating precise anomaly detection by distinguishing between benign anomalies and genuine threats [6]. This capability significantly reduces false positives, a common limitation in conventional anomaly detection strategies, thereby improving the efficiency of security operations within IoT frameworks. Safeguarding interconnected devices against sophisticated and evolving cyber threats enhances overall resilience and reliability when utilising IoT networks.

B. MLPs in the Context of IoT

MLPs, a versatile class of artificial neural networks, are particularly suited for addressing security challenges across a wide range of applications within the IoT domain, including network security [10]. Fig. 1 provides a schematic representation of an MLP, showcasing the structured arrangement of interconnected neuron layers, each linked by weights indicative of the connection strength within the network architecture.

The architecture begins with the input layer, which is responsible for receiving the initial data, where neurons $(x_1, x_2, \dots, x_n)$ are fully connected to subsequent layers through weighted connections. The core of an MLP is its hidden layers, which typically include the first hidden layer $(h_{1,1}, h_{2,1}, \dots, h_{k,1})$ and continue through to the last hidden layer $(h_{1,m}, h_{2,m}, \dots, h_{q,m})$, with each layer comprising multiple neurons that process inputs using activation functions. The output layer consists of neurons $(y_1, y_2, \dots, y_r)$, which produce the final predictions.

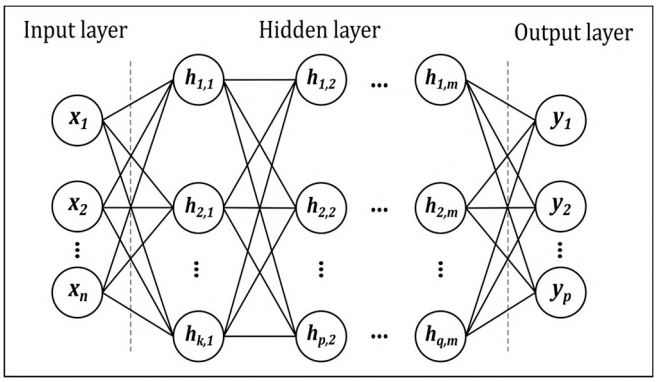


Fig. 1. *Schematic representation of an MLP*

The activation functions are critical for the model's ability to handle nonlinear data, enabling the learning of complex patterns and relationships [11]. The choice of activation functions is pivotal for the intricate needs of IoT security, which often involves binary classification. The Rectified Linear Unit (*ReLU*) (1) and *Sigmoid* (2) functions are particularly relevant. The *ReLU* function defined as

$$f(x) = max(0, x), \tag{1}$$

where x represents the input to the neuron, is predominantly utilised in the hidden layers. This preference is attributed to the *ReLU* function's efficiency in introducing nonlinearity to the model, facilitating the learning of complex data patterns [11]. Moreover, its computational simplicity arises from thresholding the activation function at zero.

The *Sigmoid* function is employed for the output layer, especially in tasks requiring binary classification [11]. The *Sigmoid* function maps the output of the final neuron to a value between 0 and 1, which can be interpreted as the probability of the instance belonging to the positive class (e.g., 'malicious'). It is expressed as

$$f(x) = \frac{1}{1+e^{-x}}, \tag{2}$$

where x is the input to the neuron. This probabilistic output is essential for distinguishing between the two classes. Consequently, the model is trained using a binary cross-entropy loss function. This function measures the divergence between the predicted probability and the true binary label (0 or 1).

Incorporating MLPs into IoT security solutions enables the detection and analysis of anomalous patterns across various devices and data points. When coupled with AutoML to optimise model selection and hyperparameter tuning, the efficiency and accuracy of anomaly detection in IoT security could be significantly enhanced. This combination can potentially improve the development of robust security mechanisms and streamline model creation, making advanced security measures more accessible.

C. AutoML in IoT

AutoML has transformed the landscape of IoT security by automating critical aspects of developing and deploying machine learning models, such as feature selection, model optimisation, and hyperparameter tuning [12]. This innovation simplifies what was once a labour-intensive and complex process, democratising access to advanced machine learning capabilities. AutoML's core strength lies in its ability to empower users to generate data-driven insights, regardless of their technical expertise, thereby enhancing the efficiency and reach of IoT security solutions [13].

Automating these phases eliminates manual oversight, reducing the time and complexity traditionally associated with customising MLP architectures and their instantiated models for specific IoT security needs.

This study implements AutoML using a custom evolutionary algorithm, which systematically evolves a population of models to find an optimal architecture and hyperparameter set. This automated search is complemented by integrating SHAP as a core interpretability mechanism, to enhance the transparency and accountability of the final model—a particularly beneficial trait in IoT security [14].

D. Interpretability in IoT Security Models

Interpretability is the capacity to elucidate and justify a model's internal mechanisms, illustrating how it processes input to arrive at decisions or predictions [15]. Interpretability is paramount within IoT security models. As models grow in complexity, particularly those rooted in deep learning, their decision-making processes may become less transparent, intensifying the critique of these models as black boxes [14].

The value of interpretability is two-fold: First, transparency and accountability become indispensable in real-world deployments of machine learning models within

the IoT domain [12]. Stakeholders must have confidence in the decision-making process of these models; thus, providing cogent explanations for the predictions made is essential for building trust and facilitating the broader acceptance of machine learning-driven decisions. Second, interpretability plays a critical role in the debugging and refinement of models. When outcomes deviate from expectations, a clear understanding of the model's reasoning can offer crucial insights for troubleshooting and enhancement, ultimately leading to improved performance and reliability [15].

Numerous methodologies have been devised to enhance interpretability in neural networks, with the application of Shapley values emerging as particularly notable [16]. Shapley values, derived from cooperative game theory, provide a systematic approach to attributing the contribution of each feature to the model's overall predictions. This technique is grounded in the fair distribution of payouts among players, representing features in the context of machine learning [17]. By calculating the marginal contribution of each feature to the prediction, Shapley values offer a transparent mechanism for understanding a model's behaviour.

The mathematical formulation of Shapley values is

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|(M-|S|-1)!}{M!} [f_x(S \cup \{i\}) - f_x(S)], \quad (3)$$

where ϕ_i denotes the Shapley value for the feature i, N represents the set of all features, S is a subset of N excluding feature i, and M is the total number of features [17]. The term $| S |$ denotes the size of the subset S, and $f_x(S)$ is the value function, such as the model's prediction, when only the features in S are considered. The factorial terms account for the permutations of feature subsets, ensuring an equitable distribution of the marginal contributions.

The proposed framework incorporates Shapley values as Deep SHAP [18]. Deep SHAP further refines this interpretability by leveraging deep learning techniques to approximate Shapley values more efficiently. Deep SHAP combines Shapley values with deep learning's capacity to model high-dimensional data, offering a scalable solution for feature attribution in complex neural networks. This advancement ensures that the interpretability mechanisms are rigorous and computationally feasible, making them suitable for the expansive and varied datasets typical in IoT environments.

While Deep SHAP provides powerful insights, calculating exact Shapley values is computationally expensive, particularly on large datasets typical of IoT environments [18]. To make this process feasible, the SHAP algorithm relies on a smaller, representative "background dataset" to approximate the model's expected output [19]. When the full training set is too large to serve as this background, a common and effective technique is to use K-Means clustering to summarise the data [17]. The resulting cluster centroids act as a condensed, weighted summary of the entire dataset. This small set of centroids is then used as the background dataset for the SHAP explainer, significantly reducing computation time while maintaining a robust approximation of the data's underlying distribution.

To leverage these SHAP values for robust feature selection, this study implements a Boruta-SHAP approach [20]. This technique is inspired by the Boruta algorithm, which provides a statistical basis for feature importance by comparing a feature's performance against randomised "shadow" features [21]. Fig. 2 provides a conceptual overview of this process.

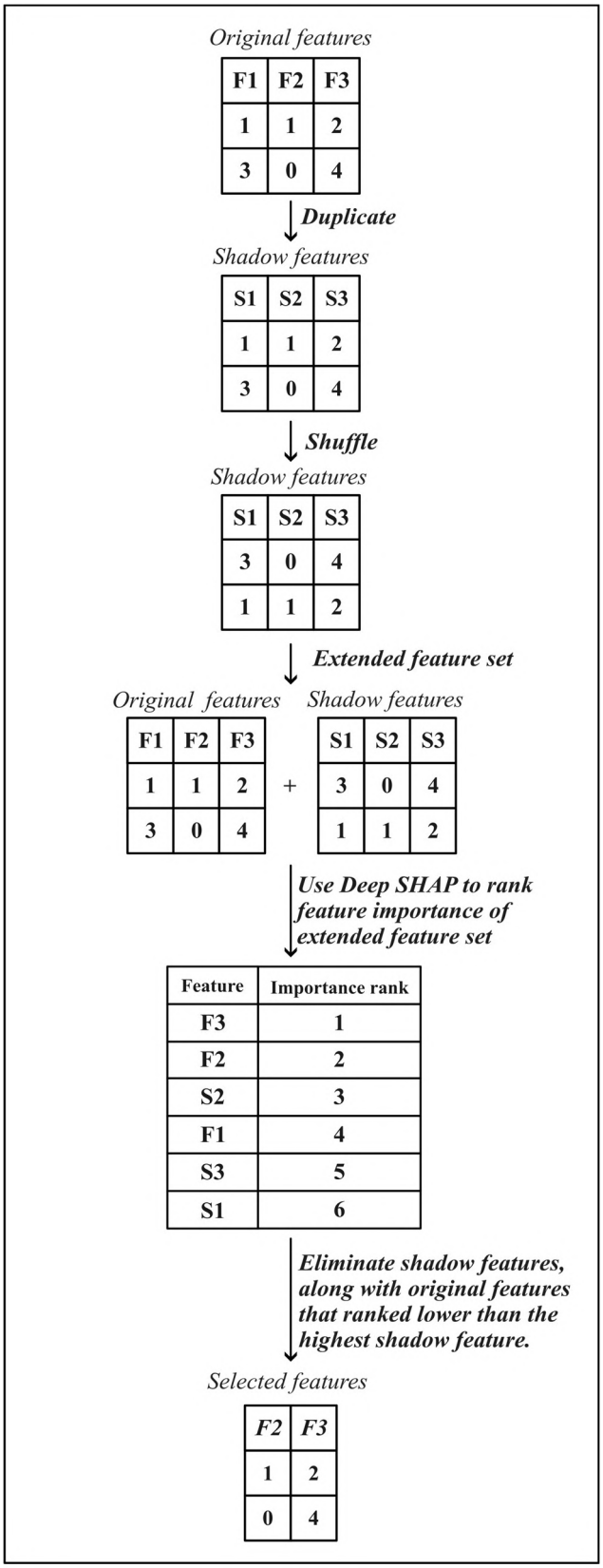


Fig. 2. *Conceptual workflow of the Boruta-SHAP feature selection process, adapted to use SHAP values as the importance metric*

A shadow feature is created by randomly shuffling the values of an original feature. This process breaks the feature's link to the target variable but preserves its statistical distribution [21]. The proposed framework uses Deep SHAP to calculate feature importance and then adopts the Boruta philosophy: a feature is only considered significant if its importance is greater than the maximum importance of any shadow feature. This method ensures that the final model is built only with features that are statistically more informative than random noise [20].

By deploying Deep SHAP within this feature selection framework and in conjunction with an MLP, the proposed model may achieve a higher level of interpretability without compromising performance. This integrated approach provides a transparent rationale for feature selection and final predictions. Furthermore, by isolating only the most relevant features, the framework may facilitate a significant reduction in dimensionality, enhancing the final model's computational efficiency and simplifying its architecture.

In the next section, the proposed framework for constructing the automated interpretable MLP architecture is outlined, detailing how these concepts are integrated to ensure the model's predictions are understandable.

III. A FRAMEWORK FOR EVOLUTIONARY AND INTERPRETABLE MLPS

The proposed framework integrates a custom evolutionary algorithm to automate the construction of an effective MLP architecture for IoT anomaly detection [22]. This evolutionary process begins with an initial population of MLP models, each with a random architecture and set of hyperparameters. Each model's fitness is evaluated within the evolutionary loop based on its validation accuracy. The fittest individuals are then selected as parents to produce offspring for the next generation through two key genetic operators: crossover, which swaps network characteristics, and mutation, which introduces random changes to elements like dropout rates, the number of layers, or the learning rate. The new generation then replaces the old, and this process iterates to progressively optimise the model's structure.

This framework employs a SHAP-Boruta feature selection stage to ensure the final model is accurate, efficient, and interpretable [20]. However, instead of using the popular Random Forest's default importance metric, the proposed approach leverages Deep SHAP for a more robust, model-agnostic measure of feature contribution [21].

The process involves creating a shadow feature for each original feature by randomly shuffling its values. The evolutionary algorithm is first trained on this augmented dataset containing both original and shadow features. To prepare for the feature contribution analysis, K-Means clustering is applied to the augmented training data, and the resulting centroids are used directly as the representative background dataset for the Deep SHAP explainer [17]. Deep SHAP values are then calculated for all features, and the maximum SHAP value from any shadow feature is used as a statistical noise threshold. Only original features with a mean SHAP value exceeding this threshold are selected for the final model training phase. This rigorous process ensures that the final MLP is trained exclusively on features proven to be more informative than random chance, simultaneously boosting performance through dimensionality reduction and simplifying the final model for better interpretability. The detailed logic of this multi-stage framework is presented in Framework 1.

Stage 1: Pre-processing

Input: Raw data

Clean data: Remove duplicates and rows with missing values.

Transform labels to binary (0 for Benign, 1 for Malicious).

Remove constant (all-zero) and highly correlated (>0.95) features.

Split data into training (64%), validation (16%), and test (20%) sets.

Normalise feature values using a MinMaxScaler fitted on the training data.

Apply SMOTE to the training set only to address class imbalance.

Stage 2: Baseline Model Training

Create an initial population of MLP models with random architectures trained on the pre-processed features from Stage 1.

Begin Evolutionary Loop (run for a fixed duration, e.g., 12 hours)

    For each individual in the population:

        Evaluate the individual's fitness using the fitness function.

    For each selected pair of parents:

        Perform crossover to generate offspring (can include swapping layers, nodes or hyperparameters between parents).

    For each offspring generated:

        Apply mutation to randomly modify the architecture or hyperparameters (can change dropout rates, number of nodes in a layer, number of layers, or the learning rate.).

    Replaces current population with selected individuals and offspring.

Select the best individual from the final generation.

Stage 3: Boruta-SHAP Feature Selection

Create shadow features by randomly shuffling each feature of the original feature set.

Augment the dataset by combining original and shadow features.

Train a single, high-performing MLP on this augmented dataset.

Create a representative background dataset by applying *K*-Means clustering to the augmented training data.

Initialise a Deep SHAP explainer with the trained MLP and the K-Means background data.

Compute SHAP values for all features (original and shadow) on test set.

Determine the noise threshold by finding the maximum mean absolute SHAP value from all shadow features.

Select the final feature set by keeping only original features with a mean absolute SHAP value greater than the noise threshold.

Stage 4: Proposed Model Training and Interpretation

Create a new, reduced dataset using only the features selected in Stage 3.

Repeat the full evolutionary process from lines **16-28**, training only on the reduced, feature-selected dataset.

Select the best-performing model from all generations as the Proposed Model.

For each prediction made by the Proposed Model:

    Compute final Deep SHAP values to understand each feature's contribution.

Return the Proposed Model, its performance metrics, and the feature-level interpretations.

Framework 1. *A Framework for an Evolutionary and Interpretable MLP*

The experimental design used to evaluate the proposed framework is detailed in the next section.

IV. Experimental design

The experimental design begins with a description of the CICIoT2023 dataset and the subsequent pre-processing steps applied to it. It then outlines the hyperparameter optimisation process used to construct the model. The section concludes with an explanation of the evaluation metrics chosen to measure the final model's performance.

A. Dataset Description

The CICIoT2023 dataset, consisting of 46,686,691 data points across 168 CSV files, is a comprehensive benchmark for intrusion detection in IoT networks [7]. The dataset was generated using a testbed that mimics a real smart home environment with 105 IoT devices. These devices include sensors, cameras, and various smart home appliances connected through Zigbee and Z-Wave protocols.

The dataset comprises 33 distinct attack types, classified into seven primary categories: DDoS, DoS, Recon, web-based, brute force, spoofing, and Mirai [7]. These attacks are characterised by 46 features detailing various aspects of IoT network traffic, including packet size, protocol type, and flag counts. Due to the significant computational resources required to process the entire dataset, this study utilises a representative subset, a common practice in related literature. The first two CSV files were used to ensure the validity of this approach and maintain comparability with the benchmark study [23]. These files, which total 457,492 data points, constitute the same sample used in the benchmark. Another key justification for this subset is its statistical representativeness. As demonstrated by [23] in Table I, the percentage distribution of the main attack categories within the sample used closely aligns with the distribution in the complete dataset.

TABLE I. Percentage distribution by attack types in the full and sample datasets

Full Dataset		**Sample Dataset**	
Attack Type	***%***	***Attack Type***	***%***
DDoS	72.36	DDoS	72.82
DoS	17.20	DoS	17.04
Mirai	5.65	Mirai	5.70
MITM	0.66	MITM	0.67
DNS	0.38	DNS	0.38
Recon	0.29	Recon	0.29
Benign	2.36	Benign	2.36

The primary traffic categories shown in Table 1, and featured in this study, are briefly described below:

- DDoS (Distributed Denial-of-Service): An attack that overwhelms a target system with traffic from multiple distributed sources, rendering it unavailable to legitimate users.
- DoS (Denial-of-Service): An attack intended to shut down a machine or network, making it inaccessible, typically originating from a single source.
- Mirai: A specific type of malware that targets IoT devices, turning them into a network of remotely controlled bots (a botnet) to launch large-scale DDoS attacks.
- MITM (Man-in-the-Middle): An eavesdropping attack where the attacker secretly intercepts and potentially alters the communication between two parties.
- DNS (Domain Name System) Attack: Involves compromising the DNS to redirect users to malicious websites or disrupt service availability.
- Recon (Reconnaissance): Consists of unauthorised network probing to gather information and identify potential vulnerabilities for future attacks.
- Benign: Represents normal, non-malicious network traffic as a baseline for identifying anomalous activities.

With the dataset's composition and key traffic types defined, the data must be prepared for model training through a series of pre-processing steps.

B. Pre-processing

Pre-processing is critical in preparing the CICIoT2023 dataset for effective machine learning model training [24]. The initial phase involved data cleaning, where data points with any missing values and duplicates were discarded to maintain dataset integrity. Following the methodology of the benchmark study [23], multi-class labels were transformed into a binary format (0 for Benign, 1 for Malicious). Additionally, constant (all-zero) and highly correlated (correlation > 0.95) features were then removed to ensure that only relevant attributes were retained. Subsequently, a sample of 457,492 data points was selected for direct comparisons with the benchmark study.

To ensure a fair and direct comparison with the benchmark study, the processed dataset was split into three subsets: 64% for training, 16% for validation, and 20% for testing. Following the split, the data was normalised using a MinMaxScaler, which transforms features to a common scale within the range [0, 1]. To prevent data leakage, the scaler was fitted only on the training data and then used to transform the validation and test sets.

A significant class imbalance exists between malicious and benign instances in the dataset. To address this, the Synthetic Minority Oversampling Technique (SMOTE) was applied. Critically, SMOTE was applied exclusively to the 64% training set after the split [23]. This strategy prevents data leakage and ensures that the validation and test sets remain representative of the real-world data distribution. By executing these pre-processing steps, the raw subset of the CICIoT2023 data was transformed into a structured format suitable for more robust machine learning model training.

Following data pre-processing, the subsequent step is to define the hyperparameter search space and the process for optimising the model itself.

C. Hyperparameter Optimisation

As established in the proposed framework (Framework 1), the optimisation of the MLP architecture is driven by a custom evolutionary algorithm. This process automates the search for an optimal model by beginning with an initial population of randomly configured MLP models. Through iterative generations, each model's fitness is evaluated based on its validation accuracy. The fittest individuals are selected as parents to produce offspring via genetic operators such as crossover, which swaps network characteristics, and mutation, which introduces random changes to elements like

dropout rates, the number of layers, or the learning rate. This evolution continues for a predefined duration, progressively optimising the model's architecture and training parameters within a bounded search space.

To guide this search, a comprehensive yet bounded hyperparameter search space was defined. The search space was determined by preliminary experimentation. To ensure a fair and direct comparison, the evolutionary search parameters detailed in Table II were held constant for the training runs of the baseline model (using all features) and the proposed model (using Boruta-SHAP selected features). The only variation was the input dimensionality of the network, which was determined by the preceding feature selection stage.

TABLE II. HYPERPARAMETER SEARCH SPACE

Parameter	Number / [Range]
Population Size	200
Parent Selection Pool Size	50
Training Duration (Hours)	12
Hidden Layers	[3, 10]
Nodes per Layer	[3, 256]
Hidden Layer Activation	ReLU
Output Layer Activation	Sigmoid
Dropout Rate	[0.2, 0.5]
Learning Rate	$[10^{-5}, 10^{-3}]$
Batch Size	[32, 4096]
Optimiser	RMSprop
Loss Function	Binary Cross-Entropy
Early Stopping Patience	15

The model produced by this automated search is then assessed using a set of standardised performance metrics to quantify its effectiveness.

D. Evaluation metrics

Evaluation metrics are essential for assessing the effectiveness of machine learning models, providing a quantitative measure of performance [25]. The following measures quantify model effectiveness: accuracy, precision, recall, F1 score, and the confusion matrix.

Accuracy, defined as

$$Accuracy = \frac{True\ Positives + T\quad Negatives}{Total\ Predictions}, \tag{4}$$

is used when the identification of true positives and true negatives is prioritised, and the impact of false positives and false negatives is less critical.

Precision is defined as

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{5}$$

and represents the ratio of true positives to total predicted positives. It is critical to minimise false positives, such as in network security, where false alarms can divert resources.

Recall, or sensitivity, is defined as

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}. \tag{6}$$

The ratio of true positives to actual positives is crucial when the cost of false negatives is high. Failing to detect anomalies in network security can lead to severe security breaches.

The F1 score, the geometric average of precision and recall, is defined as

$$F1\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Rec}. \tag{7}$$

This metric balances the predefined metrics and is helpful for an uneven class distribution. Fig. 3 depicts a confusion matrix. It provides a detailed breakdown of the model's performance, showing true positives, false negatives, false positives, and true negatives.

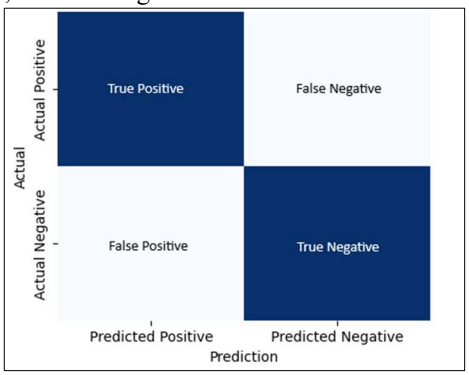


Fig. 3. *Confusion matrix*

These metrics offer a comprehensive evaluation of the model's ability to classify between different network anomalies accurately. The performance of the proposed model, as measured by these metrics, is subsequently presented and analysed.

V. RESULTS AND DISCUSSION

This section presents the empirical results of the proposed framework, evaluates its performance, and discusses the implications of the findings. To demonstrate the effectiveness of the proposed approach, two key models are examined: a baseline model developed without SHAP-based feature selection, and the proposed model, which incorporates the full interpretable framework. This section details the final optimised configurations and architectures of these models, followed by a comparative analysis of their performance against each other and against 11 established machine learning and deep learning benchmark models (as detailed in Table V). The discussion will focus on the trade-offs between accuracy, computational efficiency, and the interpretability afforded by the SHAP analysis. The core of this analysis rests on the comparison of two distinct models.

A. Selected Models

To provide a comprehensive analysis, two primary models were developed: a Baseline Model and the Proposed Model. To ensure transparency and reproducibility, the final optimised best instance of each model is detailed in Table III.

TABLE III. CONFIGURATION OF BEST MODELS

Parameter	Baseline Model	Proposed Model
Inputs	31	13
Outputs	1	1
Epochs trained	212	303
Batch size	4096	4096
Learning rate	1.0×10^{-3}	1.0×10^{-5}
MLP layers	5	9
Trial number	138	423

Furthermore, the specific layer-by-layer architectures produced by the evolutionary algorithm are presented in Table IV.

TABLE IV. FINAL ARCHITECTURES OF BEST MODELS

Layer	Baseline Model			Proposed Model		
	Nodes	*Dropout Rate*	*Activation function*	*Nodes*	*Dropout Rate*	*Activation function*
1	250	0.33	ReLU	211	0.31	ReLU
2	22	0.29	ReLU	11	0.45	ReLU
3	272	0.48	ReLU	266	0.37	ReLU
4	193	0.23	ReLU	37	0.36	ReLU
5	1	-	Sigmoid	75	0.28	ReLU
6	-	-	-	192	0.37	ReLU
7	-	-	-	192	0.41	ReLU
8	-	-	-	126	0.33	ReLU
9	-	-	-	1	-	Sigmoid

The performance of these models is then benchmarked against the deep learning models presented by [23], using the same data sample from the CICIoT2023 dataset to ensure a direct and fair comparison. All experiments were conducted for a 12-hour training period on a system equipped with an AMD Ryzen 7-5800X3D CPU, 64GB of RAM, and an NVIDIA RTX 3060 (12GB) GPU.

The first stage of this experimental process focused on creating the baseline model to establish a comparative benchmark.

B. Baseline Model Performance

The baseline model was developed by applying the evolutionary algorithm to the dataset after initial pre-processing and correlation-based feature reduction, in line with the benchmark study [23], resulting in a model trained on 31 features. The evolutionary algorithm ran for a total of 12 hours, during which it generated and evaluated 203 distinct MLP architectures. The best-performing architecture, selected based on validation accuracy, was trained for 212 epochs until early stopping was initiated, achieving a final accuracy of 99.13% and a loss of 0.0330. The total training time for this specific model was 184.33 seconds. The training history of this baseline model is depicted in Fig. 4.

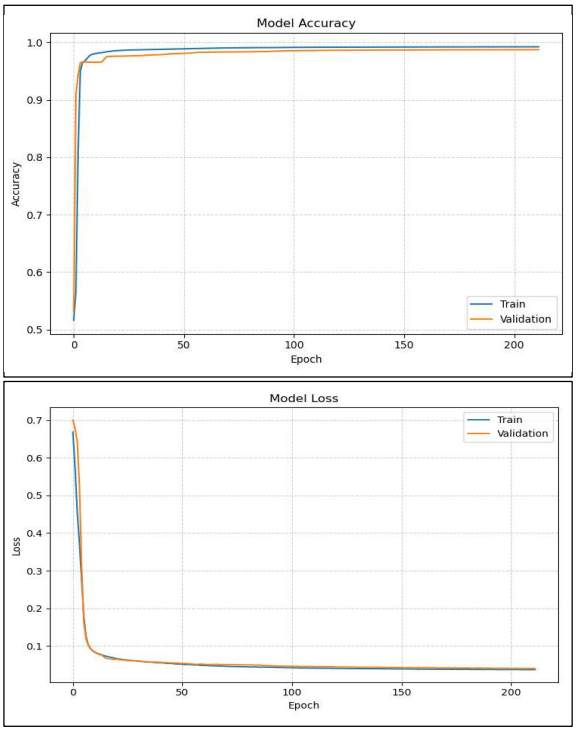


Fig. 4. *Baseline model's training accuracy and loss.*

The accuracy and loss curves show stable convergence, indicating that the evolutionary algorithm successfully identified a robust architecture.

Building on the strong performance of the baseline model, the next step in the framework is to enhance its efficiency and interpretability. This is achieved by using SHAP to identify and select only the most statistically significant features.

C. SHAP-based Feature Selection and Interpretability

A key contribution of this framework is the integration of an interpretable feature selection process. Following the Boruta-SHAP methodology shown in Fig. 2 and outlined in Framework 1, a preliminary model was trained on an augmented dataset containing the 31 original features and 31 corresponding "shadow" features. This single MLP, trained on all 62 features using the hyperparameter search space from Table II, ran for 1513.14 seconds until early stopping was triggered, achieving an accuracy of 98.23%. Deep SHAP was then used to calculate the importance of each feature from this model. Fig. 5 displays the mean absolute SHAP values for the most impactful original and shadow features. The maximum SHAP value from any shadow feature established a statistical noise threshold. Only original features with an importance value exceeding this threshold were retained. This rigorous process led to a significant dimensionality reduction of 58.06%, reducing the feature set from 31 to the following 13 most impactful features.

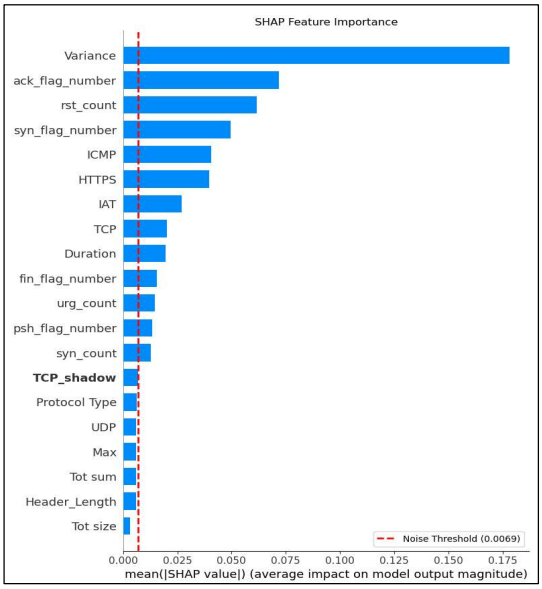


Fig. 5. *SHAP value plot with shadow features*

This step confirms that a more compact, computationally efficient, and interpretable model can be constructed using only features that are statistically more informative than random noise. With this optimised set of 13 features, the evolutionary algorithm was executed again to construct and evaluate the final proposed model.

D. Proposed Model Performance and Comparison

The final proposed model was developed by running the 12-hour evolutionary algorithm on the reduced dataset containing only the 13 SHAP-selected features. This focused approach allowed the framework to explore the hyperparameter space more efficiently, generating 696 models in the same timeframe as the baseline.

The best-performing model from this run was trained for 303 epochs until early stopping was initiated, achieving a validation accuracy of 98.55% and a validation loss of 0.0318. This resulted in a final training time of just 43.38 seconds—a 76.4% reduction in training time compared to the baseline model. The training history of the proposed model (Fig. 6) again shows excellent convergence.

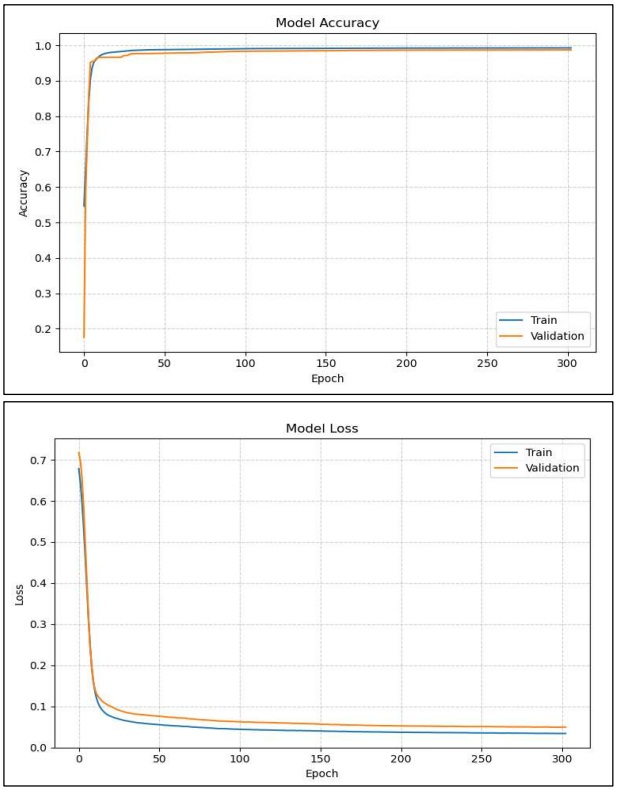


Fig. 6. *Proposed model's training accuracy and loss.*

The final evaluation metrics for the baseline and proposed models are presented in Table V, alongside results from the benchmark study [23].

The Proposed Model shows a small decrease in accuracy (-0.55%) compared to the Baseline Model. However, this is offset by a significant improvement in computational efficiency, demonstrated by the 76.4% faster training time and the 58.06% reduction in features. This efficiency gain is further highlighted by the average training time per model during the 12-hour evolutionary run. For the baseline model, the average time was 212.81 seconds across 203 models.

TABLE V. SUMMARY OF MODEL PERFORMANCE

Model	**Performance metrics**					
	Accuracy (%)	*Precision (%)*	*Recall (%)*	*F1-score (%)*	*Train time (s)*	*Test time (s)*
RF	99.60	99.92	99.67	99.79	169.54	0.478
DT	99.58	99.84	99.71	99.77	14.880	0.014
Baseline	**99.13**	**99.98**	**99.13**	**99.55**	**184.33**	**3.42**
KNN	98.89	99.94	98.92	99.43	0.089	43.681
1D-CNN	98.87	99.96	98.97	99.47	468.00	4.312
LSTM	98.87	99.87	98.97	99.42	398.03	5.692
BiLSTMv1	98.81	99.90	98.88	99.39	998.36	7.897
BiLSTMv2	98.79	99.97	98.79	99.37	784.75	7.477
BiLSTMv3	98.73	99.74	98.95	99.34	773.54	7.061
Proposed	**98.58**	**99.92**	**98.62**	**99.27**	**43.38**	**1.89**
RNN	98.31	99.98	98.29	99.13	1399.63	12.11
LR	97.90	99.98	97.87	98.91	3.649	0.006
NB	78.13	99.98	77.60	87.38	0.312	0.062

In contrast, the Proposed Model, trained on the reduced feature set, averaged only 62.07 seconds per model across 696 models. While the framework requires an initial, one-time investment to train the preliminary model (1513.14 seconds) and calculate SHAP values, this upfront cost is offset by significantly faster and more numerous model evaluations in the final evolutionary stage. This robust exploration of the hyperparameter space was achieved at a negligible cost of only 0.55% in final model accuracy. This makes the proposed model significantly more lightweight and practical for real-world deployment in resource-constrained IoT environments. This practicality is further demonstrated by the inference speed shown in Table V; the proposed model requires only 1.89 seconds for testing, representing a 44.7% reduction compared to the baseline's 3.42 seconds. Furthermore, the proposed model's final test accuracy of 98.58% remains highly competitive with the deep learning models from the benchmark study [23], while offering the distinct advantages of a fully automated, evolutionary construction process and inherent interpretability.

The confusion matrices for the baseline (Fig. 7) and proposed (Fig. 8) models confirm their strong classification performance, with few misclassifications on the test set.

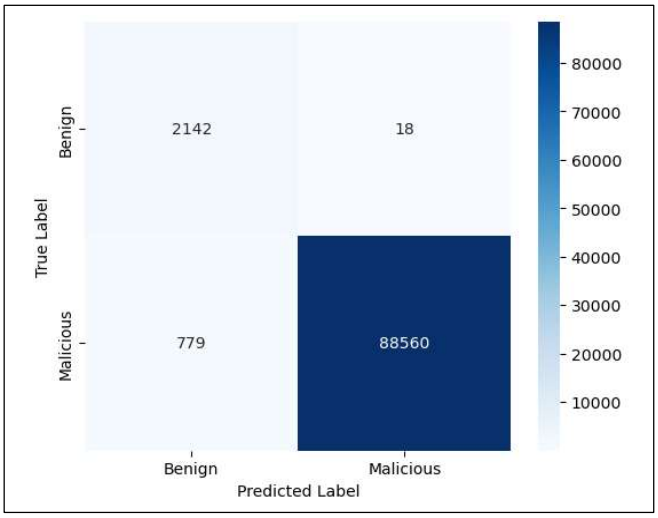


Fig. 7. *Baseline model's confusion matrix*

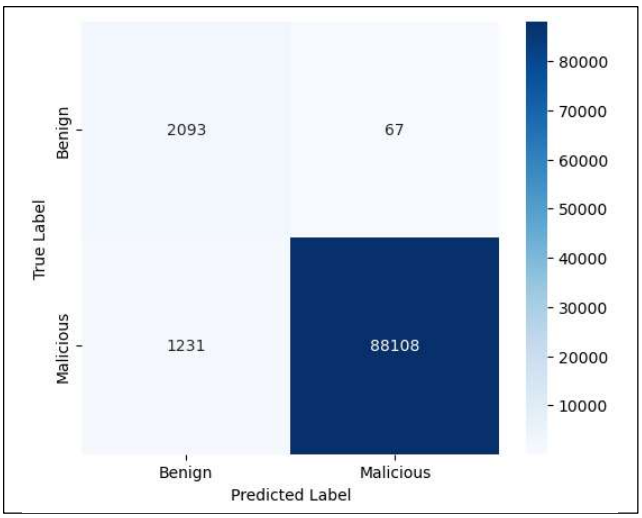


Fig. 8. *Proposed model's confusion matrix*

While these performance metrics validate the model's effectiveness, understanding why it makes its decisions is crucial for its adoption, which is the focus of the final interpretability analysis.

E. Final model interpretability

The primary advantage of the proposed framework is its transparency. The features that have the greatest impact on the model's predictions can be directly interpreted by calculating SHAP values for the final, optimised model. Fig. 9 illustrates the feature importance for the final 13-feature model.

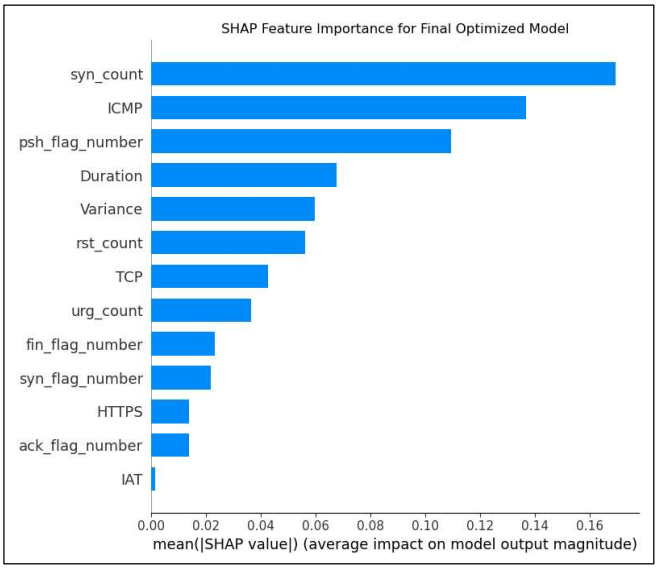


Fig. 9. *Final SHAP feature importance plot*

The plot clearly shows that features such as *syn_count*, *ICMP* and *psh_flag_number*, have the highest average impact on the model's output. These top features are directly related to network traffic behaviour commonly associated with malicious activities:

- *syn_count*: This represents the number of packets with the synchronise (SYN) flag set within a flow. A high count is a classic indicator of a TCP SYN Flood attack, where an attacker sends a succession of connection requests to overwhelm a target.
- *ICMP*: This is a binary feature indicating if the protocol is an Internet Control Message Protocol (ICMP). Since ICMP is used for diagnostics (like ping), abnormal volumes of ICMP traffic are a primary characteristic of reconnaissance scans (Ping Sweep) and certain Denial-of-Service attacks (ICMP Flood).
- *psh_flag_number*: This indicates the presence of the PUSH flag, which instructs the system to immediately push buffered data. Attackers can manipulate this flag in reconnaissance scans or to craft malicious payloads for DoS attacks.

This insight is invaluable for security analysts, as it provides a clear, data-driven explanation for why a particular network activity was flagged as anomalous. This level of transparency fosters greater trust in the AI-driven security solution and provides actionable information for network monitoring and model refinement, fulfilling the core objective of this study.

VI. CONCLUSION

This study addressed the critical need for accurate, efficient, and interpretable anomaly detection models in IoT security by introducing a framework that automates MLP design using an evolutionary algorithm and a Boruta-SHAP approach for feature selection. The empirical results demonstrated a successful balance between performance and practicality. The SHAP-based feature selection provided a 58.06% dimensionality reduction, leading to a 76.4% decrease in the final model's training time. The resulting proposed model achieved a highly competitive accuracy of 98.58% on a subset of the CICIoT2023 dataset, sacrificing only 0.55% accuracy for significant gains in efficiency and full model interpretability. The SHAP analysis also provided clear insights, confirming that features like *syn_count, ICMP* and *psh_flag_number* are critical for identifying malicious traffic. This work confirms that automating the construction of interpretable models is not only feasible but also highly effective. The proposed framework provides a robust and transparent solution that enhances trust in AI-driven security systems, paving the way for more efficient and understandable IoT protection.

REFERENCES

[1] A. Khanna and S. Kaur, "Internet of Things (IoT), Applications and challenges: A comprehensive review," *Wireless Personal Communications*, vol. 114, pp. 1687–1762, May 2020, doi: https://doi.org/10.1007/s11277-020-07446-4.

[2] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: A review," *Journal of Big Data*, vol. 6, no. 1, Dec. 2019, doi: https://doi.org/10.1186/s40537-019-0268-2.

[3] A. Assiri and H. Almagwashi, "IoT security and privacy issues," *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, pp. 1–5, Apr. 2018, doi: https://doi.org/10.1109/cais.2018.8442002.

[4] M. Bahri, F. Salutari, A. Putina, and M. Sozio, "AutoML: State of the art with a focus on anomaly detection, challenges, and research directions," *International Journal of Data Science and Analytics*, vol. 14, pp. 113–126, Feb. 2022, doi: https://doi.org/10.1007/s41060-022-00309-0.

[5] Y. Zhang, P. Tino, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 726–742, Oct. 2021, doi: https://doi.org/10.1109/tetci.2021.3100641.

[6] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security?," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 41–49, Sep. 2018, doi: https://doi.org/10.1109/msp.2018.2825478.

[7] C. Pinto, Sajjad Dadkhah, R. Ferreira, Alireza Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," *Sensors 2023*, vol. 23, no. 13, pp. 5941–5941, Jun. 2023, doi: https://doi.org/10.3390/s23135941.

[8] S. Sharmila and Rohini Nagapadma, "Quantized autoencoder (QAE) intrusion detection system for anomaly detection in resource-constrained IoT devices using RT-IoT2022 dataset," *Cybersecurity*, vol. 6, no. 1, Sep. 2023, doi: https://doi.org/10.1186/s42400-023-00178-5.

[9] M. Hasan, Md. Milon Islam, I. Islam, and M. M. A. Hashem, "Attack and Anomaly Detection in IoT Sensors in IoT Sites Using Machine Learning Approaches," *Internet of Things*, vol. 7, p. 100059, May 2019, doi: https://doi.org/10.1016/j.iot.2019.100059.

[10] B. I. Hairab, M. S. ElSayed, A. D. Jurcut, and M. A. Azer, "Anomaly detection based on CNN and Regularization techniques against zero-

day attacks in IoT networks," *IEEE Access*, vol. 10, pp. 98427–98440, 2022, doi: https://doi.org/10.1109/access.2022.3206367.

[11] D. Gangadia, "Activation Functions: Experimentation and Comparison," *2021 6th International Conference for Convergence in Technology (I2CT)*, pp. 1–6, Apr. 2021, doi: https://doi.org/10.1109/i2ct51068.2021.9417890.

[12] D. Kißkalt, A. Mayr, B. Lutz, A. Rögele, and J. Franke, "Streamlining the development of data-driven industrial applications by automated machine learning," *Procedia CIRP*, vol. 93, pp. 401–406, 2020, doi: https://doi.org/10.1016/j.procir.2020.04.009.

[13] F. Glavan and V. Croitoru, "Autoencoders and AutoML for intrusion detection," *2023 15th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–4, Jun. 2023, doi: https://doi.org/10.1109/ecai58194.2023.10194229.

[14] Z. Zhang, H. A. Hamadi, E. Damiani, C. Y. Yeun, and F. Taher, "Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research," *IEEE Access*, vol. 10, pp. 93104–93139, 2022, doi: https://doi.org/10.1109/access.2022.3204051.

[15] F.L. Fan, J. Xiong, M. Li, and G. Wang, "On Interpretability of Artificial Neural Networks: A Survey," *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 5, no. 6, pp. 741–760, Nov. 2021, doi: https://doi.org/10.1109/trpms.2021.3066428.

[16] A. Papenmeier, D. Kern, G. Englebienne, and C. Seifert, "It's Complicated: The Relationship between User Trust, Model Accuracy and Explanations in AI," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 29, no. 4, pp. 1–33, Mar. 2022, doi: https://doi.org/10.1145/3495013.

[17] A. Messalas, Y. Kanellopoulos, and C. Makris, "Model-Agnostic Interpretability with Shapley Values," *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pp. 1–7, Jul. 2019, doi: https://doi.org/10.1109/IISA.2019.8900669.

[18] P. Bhattarai, Deepa Singh Thakuri, Y. Nie, and G. B. Chand, "Explainable AI-based Deep-SHAP for mapping the multivariate relationships between regional neuroimaging biomarkers and cognition," *European Journal of Radiology*, vol. 174, pp. 111403–111403, May 2024, doi: https://doi.org/10.1016/j.ejrad.2024.111403.

[19] S. M. Lundberg and S. -I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017, vol. 30.

[20] Mehdi Jamei et al., "Monthly sodium adsorption ratio forecasting in rivers using a dual interpretable glass-box complementary intelligent system: Hybridization of ensemble TVF-EMD-VMD, Boruta-SHAP, and eXplainable GPR," *Expert Systems with Applications*, vol. 237, pp. 121512–121512, Sep. 2023, doi: https://doi.org/10.1016/j.eswa.2023.121512.

[21] G. Manikandan, B. Pragadeesh, V. Manojkumar, A. L. Karthikeyan, R. Manikandan, and A. H. Gandomi, "Classification models combined with Boruta feature selection for heart disease prediction," *Informatics in medicine unlocked*, vol. 44, pp. 101442–101442, Jan. 2024, doi: https://doi.org/10.1016/j.imu.2023.101442.

[22] J. Liang, E. Meyerson, B. Hodjat, D. Fink, K. Mutch, and R. Miikkulainen, "Evolutionary neural AutoML for deep learning," *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 401–409, Jul. 2019, doi: https://doi.org/10.1145/3321707.3321721.

[23] Ram, N.M.D.B.N., Lan, A.L.H. and Hui-Ngo, G., 2024, December. Developing Deep Learning Models to Predict Benign and Malicious Attacks in IoT Networks. *IEEE International Conference on Computing (ICOCO)* (pp. 90-95). IEEE.

[24] C. Ieracitano *et al.*, "Statistical Analysis Driven Optimized Deep Learning System for Intrusion Detection," *ArXiv*, vol. abs/1808.05633, pp. 759–769, Jul. 2018

A Comparative Evaluation of AI Models for Fraud Detection Using Real-World Malawian Banking Data

Mwayi Mwalwimba
Department of Computing Sciences
Nelson Mandela University
Qgeberha, South Africa
s228371961@mandela.ac.za

Dieter Vogts
Department of Computing Sciences
Nelson Mandela University
Qgeberha, South Africa
dieter.vogts@mandela.ac.za

Janet Wesson
Department of Computing Sciences
Nelson Mandela University
Qgeberha, South Africa
janet.wesson@mandela.ac.za

Abstract*— Fraud prevention is an essential component of risk management, which safeguards the integrity of the banking sector and client banking operations. As fraud increases in complexity, machine learning (ML) emerges as a transformative method that enhances fraud detection processes. However, the performance of ML models is negatively impacted by class-imbalanced datasets, where traditional evaluation metrics like accuracy focus on the majority class in the dataset. This study investigates the application of six ML models: Random Forest, Logistic Regression, Decision Tree, Extreme Gradient Boosting, Light Gradient Boosting Machine and Categorical Boosting. These models were trained on an imbalanced real-world banking dataset, containing 1.8% fraudulent transactions. To handle the inherent challenge of class imbalance, Random Oversampling, Random Undersampling, the Synthetic Minority Oversampling Technique (SMOTE), and three SMOTE variants, including SMOTEENN, SMOTETomek, and Borderline-SMOTE resampling methods, were employed. The model performance was evaluated through 5-fold cross-validation in terms of precision-recall area under the curve (PR-AUC) metrics to better capture the operational trade-offs in fraud detection. The results show that XGBoost trained on SMOTE resampled data achieved the highest performance, with a PR-AUC of 0.2225, precision of 0.1299, and recall of 0.4667. While the performance score of PR-AUC 0.2225 is lower than values often reported on synthetic data, these results reflect the inherent difficulty of fraud detection under real-world conditions. Furthermore, the study demonstrates the critical role of class imbalance handling techniques in enhancing ML model performance in fraud detection.

Keywords—Artificial Intelligence, Classification Models, Machine Learning, Imbalanced data, Resampling Techniques, Banking Sector, Financial Data, Banking Fraud Detection

I. INTRODUCTION

Financial fraud is an increasing threat to the operation and integrity of the banking sector. As fraud increases, traditional rule-based fraud detection methods have shown deficits in identifying patterns of fraudulent transactions [1]. As a result of limitations of traditional methods, several advanced fraud detection methods have been applied in the banking sector. Machine Learning (ML) methods present a transformative shift in financial fraud detection through accuracy and effectiveness [2], [3], [4]. ML models can identify actual fraudulent patterns with minimal false predictions, which is crucial to maintain integrity and prevent fraudsters from using the banks as an instrument to conduct financial crimes.

Despite the advancements in ML for fraud detection, data class imbalance presents critical implementation challenges [5], [6].In the banking dataset, class imbalance is a state where legitimate transactions occur more regularly than fraudulent transactions [7]. Class imbalance affects the class distribution in the dataset and has a bearing on the model's performance, especially when applying classification methods for fraud detection [5], [7]. Therefore, in fraud detection, where accurate distinction of legitimate and fraudulent patterns is vital, a model aligned with the majority may lead to overlooked potential fraud, allowing fraudulent transactions to go undetected [8]. It is essential to use an appropriate evaluation metric to assess the model's performance under imbalanced data, since accuracy may not be meaningful in a highly imbalanced dataset, as the model fails to learn and distinguish between majority and minority classes [9], causing the model to align with the majority class, hence presenting biased results, for instance, illegal financial activities being considered legal [5], [9], [10], [11].

To effectively evaluate the performance of models in an imbalanced dataset, Tanha et al. [13] and Wasswa et al. [14] recommended using multiple evaluation metrics as they can ensure comprehensive model evaluation, thereby handling the problem of dependence on class distribution in the dataset [12], [13]. According to Wasswa et al. [14], metrics such as precision-recall area under the curve (PR-AUC), area under the receiver operating characteristic (ROC) curve, and the F1-score have been proposed to be effective in classification tasks with imbalanced data.

The existing literature on ML for fraud detection primarily focuses on synthetic datasets and real-world banking data from developed economies. However, few studies have explored real-world datasets from Sub-Saharan African countries, despite the region's growing digital banking sector and unique fraud risk landscape. Fraud differs across regions, so the methods to detect it may not be directly applied across the board. As a result, there is a need for fraud detection methods that were trained on and use local data to ensure that the results reflect real-world conditions in the Sub-Saharan region. While the data may not directly model performance metrics, it enhances the practical relevance and interpretability of the findings; hence, this study will contribute to the field of fraud detection in Malawi.

This study investigates the effect of class imbalance handling methods on the performance of Random Forest (RF), Logistic Regression (LR), Decision Tree (DT), Light Gradient Boosting Machine (LightGBM), Extreme Gradient Boosting (XGBoost), and Categorical Boosting (CatBoost) ML models. The study involves comparative experiments to examine how

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

Synthetics Minority Over-Sampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), SMOTE TOMEK, and Borderline_SMOTE class imbalance handling methods influence models' recall and precision under imbalanced real-world banking transaction data. The study aims to identify the optimal model configuration, hyperparameter tuning, feature engineering techniques, and optimisation methods that can establish an acceptable standard for practical deployment in the real world, balancing fraud detection with minimal false positives.

The rest of the paper is organised as follows: Section II presents the background and the data collection procedure. Section III reviews the literature on fraud detection, class imbalance and model evaluation metrics. Section IV presents ML concepts and algorithms used in this study. Section V describes the methodology, focusing on study design and the dataset's exploratory data analysis. Section VI presents the experimental procedure, which includes model training and evaluation. Section VII presents results and discussion. Section VIII is the conclusion, and Section IX is the acknowledgements. Finally, the paper contains a list of references.

II. Background And Data Collection

This section introduces the Malawian banking sector and describes the data collection procedure.

A. Overview of the Malawian banking sector

The Malawian banking sector comprises eight banks. All eight banks have a branch network of 120 branches, 95 kiosks and 498 Automated Teller Machines (ATMs) across Malawi. The banking sector is experiencing an increase in the use of payment products and systems, driven mainly by the adoption of Digital Financial Services (DFS) [15]. In 2024, interbank transfer was a key contributor to the overall volume and value of transactions processed through the National Payment System (NPS). Eight million interbank transactions were processed with a corresponding value of MWK 128.3 trillion. As the banking sector is adopting digital systems and offering digital services, the volume of cash withdrawal transactions has decreased, with only 39,498,758 transactions with a corresponding value of MWK 20,893.9 [16]. According to the Financial Intelligence Authority (FIA), an authority responsible for analysing financial information and producing financial intelligence to combat financial crimes, funds transfer fraud is a prevalent risk in the banking sector [16].

B. Data Collection

The study adopted a quantitative research design and utilised a real-world banking transactions dataset. The Nelson Mandela University Research Ethics Committee granted ethical approval for the study, and additional clearance was obtained from the National Commission for Science and Technology (NCST) in Malawi, a national-level ethics body. Furthermore, a formal written consent to access the banking transaction dataset was obtained from the Financial Intelligence Authority (FIA).

III. Related Work

Section III presented existing research studies on applying ML algorithms for fraud detection. This section further highlights diverse class imbalance handling methods in the banking transaction datasets.

The dynamic nature of financial fraud has always been a growing problem in the banking sector, making fraud detection a critical component of the banks' risk management strategy. Although banks have done extensive work to implement fraud detection systems, the systems remain vulnerable to fraudsters because they lack a holistic, 360-degree view of transaction information [18]. According to Hashemi et al. [19], fraudsters learn how the systems work and use the vulnerabilities to use banks as instruments to commit financial crimes. The situation has made fraud detection complicated, necessitating banks to implement advanced technological systems in transaction monitoring to detect and prevent fraud. Researchers are constantly working on advanced methods to improve the performance of existing fraud detection and prevention systems.

A range of ML algorithms, from baseline models to complex ensembles, have been applied in transactions to detect and prevent fraud [20], [21]. Despite ML's potential in fraud detection, the class imbalance problem limits model performance, making it difficult for algorithms to learn predictive patterns from highly imbalanced data.

To deal with class imbalance in real-world data and its effects on accurate model classification, researchers have recommended applying data-level and algorithm-level methods [22]. Data-level imbalance handling methods help arrive at a complete, balanced dataset for the ML model process to produce accurate results [9]. According to Carvalho [23], data-level approaches involve modifying the data distribution in the dataset to amplify the relative importance of the minority class during model training. In comparison, algorithm-level methods for handling class imbalance involve modifying the learning process of the algorithm to increase the importance of the positive class [24].

To improve the performance of fraud detection in a class-imbalanced dataset, Isangediok and Gajamannage [25] investigated the performance of LR, DT, RF and XGBoost by training the models on data resampled using diverse resampling strategies: SMOTE, Random Undersampling (RUS) and SMOTE and Edited Nearest Neighbours (SMOTEENN), to maximise recall and reduce false negatives. Even though RandomisedSearchCV was used for hyperparameter tuning, XGBoost trained on the original imbalance dataset outperformed LR, DT and RF in terms of detecting actual fraudulent patterns as fraud. Hashemi et al. [19] utilised a class weight-tuning hyperparameter to control the weight of classes in the dataset. The study also employs an ensemble voting approach that combines CatBoost and XGBoost to enhance the prediction performance of LightGBM on real-world data. Deep learning and Bayesian optimisation methods were utilised for hyperparameter optimisation to improve the model's performance. Cross-validation was applied to the ML algorithms. The results of the study demonstrated that ensemble learning approach has the potential to improve prediction as the performance improved as follows: from Receiver Operating Characteristics, Area Under the Curve (ROC-AUC) = 0.95, precision 0.79, recall 0.80, F1 score 0.79, and Matthew's correlation Coefficient (MCC) 0.79, to ROC-AUC = 0.94, precision = 0.80, recall = 0.82, F1 score = 0.81, and MCC = 0.81.

Contrary to the study by Isangediok and Gajamannage [25], the results of a study by Detthamrong [26] demonstrated that applying a data-level class imbalance handling approach significantly improved fraud detection. The study employed diverse sampling, scaling methods with voting ensemble learning methods and multiple ML models: CatBoost, LightGBM, XGBoost, LR and Neural Network (NN). From the results, CatBoost performed well with a precision of 0.9756, a recall of 0.8163, an F1 score of 0.8889 and a ROC-AUC of 0.9081.

Subsequently, Sariat et al. [27] investigated the application of soft voting ensembles, utilising XGBoost, LR, and RF. The models were trained on synthetic data resampled using SMOTE to handle class imbalance. The study conducted comprehensive feature engineering by introducing error calculation mechanisms. The results of the study demonstrated that resampling strategies coupled with feature engineering prove to be effective in improving fraud detection accuracy of ML algorithms. RF has a Precision-Recall Area Under Curve (PR-AUC) of 0.9998. A similar study was done by Kaushik et al. [28], where a combination of no sampling, random undersampling, random oversampling, SMOTE, and Tomek link removal data resampling methods was applied to ensemble classifiers XGBoost, LightGBM and Random Forest to detect fraud in imbalanced data. To compare the model's performance, the researcher employed multiple evaluation metrics: precision, recall, F1 score, MCC, PR-AUC curve and ROC-AUC. Individual models were assessed based on the overall evaluation metrics, and Random Forest with SMOTE Tomek-like removal outperformed XGBoost and LightGBM.

The review of the related research studies has demonstrated that in ML modelling for fraud detection, there is no one-size-fits-all solution. As the size of banking transactions and the complexities of financial fraud continue to increase in the banking sector, there is a need for continuous research to refine and advance fraud detection methods [27].

IV. Basis Of Machine Learning Concepts And Algorithms

This section presents the ML models used in this study.

A. Machine Learning Algorithms

The study incorporated different ML models to enhance the robustness of fraud prediction models.

1) Decision Tree (DT) is a non-parametric supervised learning algorithm for classification and regression tasks [29]. DT was selected as a base model due to its ability to classify data elements and capture non-linear, complex relationship patterns in the data with minimal effort.

2) Random Forest (RF) is an ensemble ML technique that combines multiple models to build a powerful model for classification and regression problems. RF was adopted due to its ability to accurately capture diverse and unusual patterns in the data [30] and weigh them appropriately for accurate predictions, which is important for fraud detection. RF can handle large datasets with high dimensionality and noisy and complex interactions among features [31]. Furthermore, their ease of interpretation makes them suitable for this study, allowing for an interpretable and robust fraud detection model.

3) Logistic Regression (LR) is a predictive analysis algorithm that analyses if two or more variables are related. LR was selected because they are effective for binary and multiclass classification tasks and highly scalable to work with large datasets [30]. Furthermore, LR is fast in data processing and easy to calibrate for good performance and interpretation, making it a valuable model for fraud detection[32].

4) Gradient Boosting is an ensemble method that structures models sequentially, and each model tries to improve the training speed and generalisation capability [33]. XGBoost, LightGBM and CatBoost were incorporated to boost the model's performance for fraud detection. Gradient boosting algorithms can handle structured datasets with high performance [34], capturing intricate patterns and relationships within the data, leading to high accuracy in classification

a) XGBoost was applied due to its scalability, speed and ability to analyse large imbalanced datasets effectively. XGBoost is robust in handling monotonic nonlinear relationships within the data. Furthermore, XGBoost can handle missing values and outliers in the dataset to enhance prediction accuracy [29], [35],[36], which is crucial for fraud detection tasks.

b) LightGBM is a distributed gradient boosting framework which utilises decision trees as base models [37]. LightGBM was used due to its efficiency in handling large-scale and structured datasets [38], which is ideal for the dataset used in this study.

c) CatBoost is a gradient boosting ML algorithm focusing on categorical variables [39]. The study adopted CatBoost because of the model's ability to handle categorical and ordered features without requiring considerable pre-processing [40]. CatBoost requires less training to capture intricate patterns and relationships within the dataset. Furthermore, Catboost can be applied to diverse data types [41], which makes the algorithm ideal for fraud detection in real-world datasets.

B. Class Imbalance Handling Methods

Synthetic Minority Oversampling Techniques (SMOTE), Adaptive Synthetic Sampling (ADASYN), Class weight, SMOTE_TOMEK and Borderline SMOTE resampling techniques were applied on the training set. The methods created synthetic new samples by interpolating between minority class instances that connect via line segments to tie nearest neighbours within the same class [22].

C. Model Evaluation Metrics

In classification, multiple evaluation metrics can be applied to evaluate the performance of ML models. The metrics include (a) Precision, (b) F1-score, (c) recall, (d) accuracy, (e) MCC and confusion matrix. In a confusion matrix, True Positive (TP) represents fraudulent transactions that have been classified as fraudulent, False Positive (FP) represent legitimate transactions misclassified as fraudulent, True Negative (TN) represents legitimate transactions correctly classified as legitimate, and False Negative (FN) represent fraudulent transactions which

were wrongly classified as legitimate. Table I represents a summary of mathematical equations for the evaluation metrics:

TABLE I. PERFORMANCE METRICS AND FORMULAE

Evaluation Metric	Formula
Accuracy	(TP + TN) / (TP + TN + FP + FN)
Precision	TP / (TP + FP)
Recall	TP / (TP + FN)
F1 Score	2 * TP / (2 * TP + FP + FN)
MCC	TP * TN - FP * FN) / sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))

D. Experimental Setup

The experiments were conducted in a Python programming environment. The Scikit-learn (sklearn) ML libraries were utilised for ML model implementation. All experiments were performed on a machine with a 13th Generation Intel Core ™ i7 processor, featuring eight cores at 2.10 GHz and 32.0 GB of RAM.

V. METHODOLOGY

This study conducted a comparative experiment by applying ML models to real-world banking transaction data for fraud detection. As part of the study, the following methodological workflow was adopted:

A. Exploratory Data Analysis

ML models were trained and evaluated on a real-world banking transactions dataset from the Malawian banking sector. The dataset consists of 114,735 transaction records made between 1 January 2022 and 31 December 2022. The dataset includes 2026 (1.8%) transactions labelled as fraudulent, which were identified based on the financial crime risk indicators used in the banking sector. The dataset has nine features, out of which two features, namely; initiating and recipient accounts, were anonymised. Table II presents the features in the dataset.

TABLE II. FEATURE DESCRIPTION

Feature Name	Description
Transaction Date	The day the transaction took place.
Initiating Account	Unique number identifying the initiating account
Description	Narration of the transaction
Recipient Account	Unique number identifying the recipient account
Transaction Type	Categorical variable indicating the type of transaction
Currency	Categorical variable indicating the type of currency
Transaction Amount	Amount transacted
IsFlaggedFraud	A binary variable indicates transactions flagged as fraudulent but not confirmed as fraudulent.
IsFraud	A binary variable indicating if the transaction is fraudulent or legitimate (0 = legitimate, 1 = fraudulent)

To understand the distribution and structure of the raw real-world dataset, the EDA further presents the transaction class distribution in Fig. 1, financial transaction network analysis.

Data Class Distribution: The dataset shows a class imbalance with 114,746 transactions, of which only 2026 are confirmed as fraudulent, representing 1.8 %. Fig. 1 illustrates class distribution in the dataset.

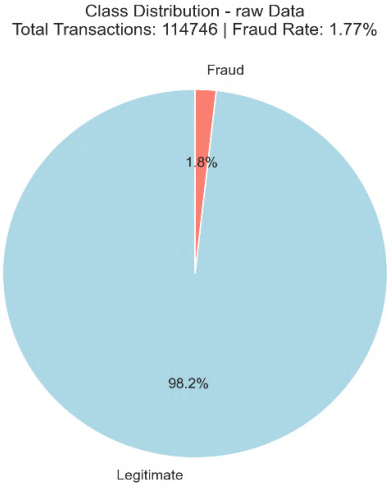


Fig. 1. Class Distribution

a) Graph Network Analysis: The Graph Network Analysis in Fig. 2 represent a financial transaction network in the dataset. Red dots present fraudulent transactions, while green dots represent legitimate transactions. The grey arrow represents the transaction flow from the initiating account to the recipient account(s).

To understand the dataset and the relationships between legitimate and fraudulent transaction patterns, Graph Network Analysis was employed to visualise the transaction flow network and quantify pairwise feature relationships.

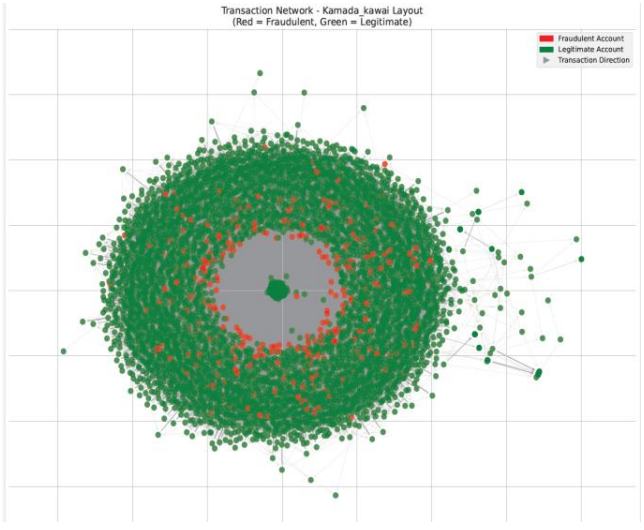


Fig. 2. Financial Transaction Network Analysis

The EDA reveals that the network of transactions is dense and has highly connected transactions. Legitimate transactions

are the majority and have a large cluster in the centre. Fraudulent transactions are the minority clustered closer to the core of the transaction network, where accounts are highly connected. The Graph Network Analysis reveals that fraudulent accounts are not isolated but embedded within the legitimate transactions network. Hence, network-based features such as topological properties, centrality indices, clustering coefficient, and neighbourhood composition could help identify patterns of legitimate and fraudulent transactions in the dataset.

B. Data Pre-processing

To have data with suitable quality for ML modelling, the following data pre-processing tasks were carried out:

a) Handling Missing Values: The raw dataset had five missing values, which were handled by the imputation method, using the mean of the respective attribute.

b) Feature Selection: The "description" column was dropped from the raw dataset because the feature had a weak correlation with the target features and contained textual data, which is unsuitable for classical ML algorithms.

C. Feature Engineering

In fraud detection, constant changes in fraudulent transaction patterns affect the performance of the ML model [27]. The study introduced new features in the training set based on the transaction behaviour to enhance how the ML model distinguishes patterns within the dataset.

a) Feature Extraction: The raw dataset has a transaction date feature composed of date and time. While the date could be a good predictor of fraud, six more features were extracted from the transaction data to enrich it, as detailed below:

"Transaction_Day", "Transaction_Month",
"Transaction_Year", "Transaction_Hour",
"Transaction_Minute", "Transaction_Second".

The additional features enhanced the dataset by introducing numerical features that gave insight, which was applied in the classification fraud detection task.

b) Handling Categorical Features: Categorical features "transaction_type" and "currency" were converted into numerical variables by using the One-Hot encoding method. The process ensured that all features were in a format that ML algorithms can process.

c) Domain Specific Feature: According to Baesens et al. [12], adding domain or expert knowledge into feature engineering can enhance the predictive performance of ML models. New features were created based on commonly used risk indicators to handle the problem of non-linearity between features in the dataset and enhance model performance. Creating new features ensured the dataset captured expert-informed characteristics of legitimate and fraudulent transaction patterns. Engineered behavioral and amount level features: "amount_pct_change_1d", "time_since_last", "txn_count_24h", "cust_max_amount", "amount_vs_amount", "is_prompt_transfer"

D. Feature Correlation Analysis

Pearson's Coefficient Matrix was employed to visualise and quantify pairwise feature relationships to understand the dataset and the relationships between features and handle multicollinearity issues.

The observation of the feature engineering process shows that the dimension of the features increased from 9 in the original dataset to 18 features. There was a positive correlation between the 13 pairs:

- transaction amount and cust_max_amount are moderately correlated (≈ 0.30). Customers with higher historical maximum transactions are more likely to have fraudulent transactions.
- transaction_type_transfer and IsFraud are highly correlated (≈ 0.33).
- amount_pct_change_1d is moderately linked to fraud with (0.28), which reflects that domain-specific feature change in transaction patterns provides a moderate pattern for fraud detection.
- Transaction_month has a very weak effect on fraud with (0.14).

Most features in the dataset have weak correlation with the target variable "IsFraud", which demonstrates non-linear feature interactions (patterns). Transaction type is one of the stronger fraud indicators in the dataset, while transaction size and customer historical behaviour, depicted by customer maximum amount and amount change per day, provide moderate fraud signals.

Feature engineering introduced new features to the dataset; therefore, to ensure model stability and increased performance we adopted approach proposed by [46], where Principal Component Analysis (PCA) was combined with Tree-based ML models to enhance fraud detection. Furthermore, PCA was applied for dimensionality reduction to avoid multicollinearity and redundancy in the training set, while preserving critical information regarding banking transaction behaviour, which is vital for fraud detection tasks [42].

Furthermore, tree-based ensemble models were employed to address the weak correlation between features and the target variable "IsFraud" and the non-linearity within the dataset to provide better interpretability and efficiency in fraud detection. The selected features facilitated the model to distinguish between legitimate and fraudulent transaction patterns.

VI. MODEL TRAINING AND EVALUATION

This section describes the implementation of a comparative experimental study, which involved model training, data sampling, hyperparameter tuning, and model evaluation.

A. Model Training

A stratified train-test split was used to divide the data into an 80% training set and a 20% testing set. The training set was allocated 80% of the data to allow robust training while maintaining enough samples for evaluating the model's performance.

This study employed stratification to ensure that each cross-validation fold retained a proportionate representation of class distributions [43].

B. Data Resampling

The raw dataset had a significant class imbalance, where 98.2% were legitimate transactions and only 1.8% were fraudulent transactions. To handle the class imbalance in the dataset, oversampling methods were applied to increase the minority class's representation. Oversampling methods were utilised because they preserve essential information contained in the minority class instances, which is ideal for fraud detection, as each transaction record holds a significant value for identifying fraudulent behaviour. Furthermore, the methods can introduce a higher degree of diversity into the minority class, which the model can use to generate the results [44], the oversampling method. Each ML model was evaluated for comparison purposes to determine how class imbalance handling methods influenced performance. All models were trained on the original training dataset without any sampling to compare the performance of the models. Table III summarises the data resampling process.

TABLE III. SUMMARY OF DATA RESAMPLING

Resampling Strategies	**Dataset after Train-Test Split**		**Resampled Training Set**	
Resampling Method	**Legitimate**	**Fraud**	**Legitimate**	**Fraud**
None	90175	1621	90175	1621
ADASYN	90175	1621	90175	29357
SMOTE	90175	1621	90175	29757
SMOTE_TOMEK	90175	1621	86691	26273
BorderlineSMOTE	90175	1621	90175	29757

C. Selected Models and Hyperparameter Tuning

Table IV outlines the ML models and summarises the hyperparameter tuning.

TABLE IV. PARAMETER SETTINGS PER MODEL

Model	**Parameter Values**
LR	'Max_iter=1000, random_state=42, uniform(0.001,10)
DT	'Random_state=42, max_depth': randint(3, 30), min_samples_split': randint(2, 20)
RF	'random_state=42, n_jobs=-1, 'n_estimators': randint(50, 300), 'max_depth': [None] + list (range (5, 30, 5)), 'min_samples_split': randint(2, 20),
XGBoost	'eval_metric='logloss', random_state=42, n_estimators': randint(50, 300), max_depth': randint(3, 10), learning_rate': uniform(0.01, 0.3), scale_pos_weight': [counts[0]/counts[1], 1], gamma': uniform(0, 5)
LightGBM	'random_state=42, n_estimators': randint(50, 300), 'num_leaves': randint(20, 150), max_depth': randint(3, 15), class_weight': ['balanced', None], 'min_child_samples': randint(20, 500)
CatBoost	'random_state=42, iterations': randint(100, 500), depth': 'randint(4, 10), learning_rate': uniform(0.01, 0.3), 'auto_class_weights': ['Balanced', 'None'], l2_leaf_reg': 'uniform(1, 10)

From Table IV, the models' hyperparameters were optimised using Randomised Search Cross-Validation. Optimal hyperparameter tuning for each model was randomly searched over the specified hyperparameter range with cross-validation. Randomised search of hyperparameters was used to set optimal parameters for each model. Randomised search was adopted because fraud detection requires systems that operate in less time. Randomised search can find hyperparameter values close to optimal values and yield better prediction results in less time [45]. The dataset was partitioned by a stratified split into five equal-sized validation pairs to have a robust model. The training model had four subsets, while the validation models had one subset. The validation process is essential in fraud detection to ensure the generalisation of the model's performance. To balance the performance of each model and the effect of randomness, interaction was set to 100. Table IV presents a list of parameter setting values for the proposed models. Model Evaluation Metrics

Precision, recall, F1 score, PR-AUC and ROC evaluation metrics were applied to evaluate the performance of the six models. Since the focus of the study was to maximise true positives and minimise the false positives in fraud prediction, PR-AUC was applied as the primary evaluation metric.

VII. RESULTS AND DISCUSSION

In this study, efficient models were assessed based on their ability to maximise true positives and minimise false positives; therefore, PR-AUC was employed to evaluate the performance of the six models. Furthermore, PR-AUC was selected to compare model performance on a highly imbalanced real-world dataset, as it emphasises the prediction of the minority class without being influenced by the majority class.

A. Model Performance Evaluation Overview

The comparative experiments of all the combined modelling approaches in Table V revealed variation in fraud detection performance across the six models and resampling strategies. The results demonstrated the challenge of fraud detection in a real-world banking dataset due to high class imbalance. The class imbalance affected recall when ML models were applied without resampling the training set, despite high accuracy scores, which were driven by most of the legitimate transactions class. Class imbalance handling methods, such as SMOTE, ADASYN, Borderline-SMOTE, and SMOTE-Tomek, enhanced the predictive ability of most models by improving recall at the cost of precision. However, the results have shown that the degree of improvement and the stability of the precision–recall trade-off varied substantially across different algorithms.

Overall, XGBoost demonstrated better PR-AUC performance than other gradient boosting methods CatBoost and LightGBM. Table V presents an overview of the key model performance evaluation metrics across different class imbalance handling strategies.

TABLE V. PERFORMANCE EVALUATION METRICS FOR SIX ML MODELS IN FRAUD DETECTION

ML Model	Resampling	ROC_AUC	PR_AUC	F1	Recall	Precision
CatBoost	SMOTE	**0.8326**	0.2142	0.1919	0.4395	0.1228
	SMOTE_TOMEK	0.8299	0.1917	0.1895	0.4444	0.1204
	Borderline_SMOTE	0.8063	0.1762	0.1848	0.4346	0.1173
	ADASYN	0.8278	0.1556	0.1779	0.4741	0.1095
	None	0.8116	0.1244	0.0000	0.0000	0.0000
Decision Tree	None	0.6779	0.1066	**0.2475**	0.2148	0.2919
	Borderline_SMOTE	0.7308	0.0823	0.1816	0.4593	0.1131
	SMOTE	0.7373	0.0613	0.1601	0.4370	0.0980
	SMOTE_TOMEK	0.7256	0.0611	0.1491	0.4938	0.0878
	ADASYN	0.7384	0.0597	0.1518	0.4543	0.0911
LightGBM	SMOTE	0.8204	0.2007	0.2190	0.3778	0.1542
	SMOTE_TOMEK	0.8235	0.1641	0.1030	**0.6988**	0.0556
	ADASYN	0.8224	0.1432	0.1983	0.4099	0.1308
	Borderline_SMOTE	0.8012	0.1353	0.1175	0.6123	0.0650
	None	0.8025	0.1207	0.0288	0.0148	**0.5455**
Logistic Regression	SMOTE	0.5993	0.0308	0.0572	0.2074	0.0332
	None	0.5990	0.0307	0.0574	0.2074	0.0333
	Borderline_SMOTE	0.5892	0.0307	0.0526	0.3012	0.0288
	ADASYN	0.5992	0.0306	0.0546	0.2074	0.0314
	SMOTE_TOMEK	0.5943	0.0305	0.0536	0.2321	0.0303
Random Forest	None	0.7560	0.1923	0.2825	0.2173	0.4037
	SMOTE	0.7872	0.1400	0.1771	0.4272	0.1117
	SMOTE_TOMEK	0.7649	0.1393	0.1525	0.4568	0.0915
	Borderline_SMOTE	0.7738	0.1297	0.1867	0.4370	0.1187
	ADASYN	0.7783	0.1138	0.1660	0.4568	0.1014
XGBoost	SMOTE	0.8287	**0.2225**	0.2032	0.4667	**0.1299**
	SMOTE_TOMEK	0.8206	0.1971	0.1848	0.4321	0.1175
	Borderline_SMOTE	0.8078	0.1628	0.1885	0.4099	0.1224
	ADASYN	0.8216	0.1516	0.1718	0.5037	0.1036
	None	0.8078	0.1226	0.0098	0.0049	0.6667

a) Precision Recall – AUC Comparison

Table V. Presents PR-AUC values of overall model performance. Among the proposed six models, LR remained near trivial levels (~0.03) regardless of resampling. LR had the lowest PR-AUC value of 0.0308, trained on the data resample with SMOTE. The results indicated that LR failed to prioritise fraud effectively under a class-imbalanced dataset, regardless of resampling. Although DT improved recall over LR, they achieved a PR-AUC value of 0.0597 when trained on data resampled with ADASYN. DT was aligned towards the majority class in the imbalance dataset and failed to detect fraudulent transactions accurately. RF improved upon this baseline, with PR-AUC around 0.1923, trained on non-sampled data. However, the PR-AUC value dropped when the model was trained on oversampled data, suggesting some sensitivity to noise, which was introduced by synthetic samples.

In contrast, the comparative analysis showed that ensemble methods, especially gradient boosting methods, yielded better results by balancing between false positives and missed predictions. XGBoost resampled with SMOTE achieved the highest performance: PR-AUC of 0.2225, Precision of 0.1299 and Recall of 0.4667, which demonstrated the model's ability to identify the highest percentage of the fraudulent patterns without being influenced by the legitimate patterns. CatBoost with SMOTE was the second best-performing model with a PR-AUC of 0.2007. LightGBM with SMOTE managed to optimise the trade-off between recall and precision with PR-AUC values of 0.1083 trained on resampled data, and 0.1454 with SMOTE.

The results of the study are significant because they demonstrate the model's ability to achieve high recall while maintaining reasonable precision levels across a range of thresholds. The resampling strategies improved the model's ability to detect fraud without minimising precision. Furthermore, XGBoost and CatBoost performed better due to their ability to iteratively refine predictions on fraud instances within the dataset, achieving results that allowed flexible threshold selection for operational needs. The overall performance of the ensemble with SMOTE suggested inherent dataset class imbalance challenges.

b) ROC Curves Comparison

The comparative experiment also analysed how models can distinguish between legitimate and fraudulent transactions in the real-world dataset. The ROC curve values in Table VI illustrate that models have varying performance results. Among the six models, XGBoost, LightGBM, and CatBoost are the best performers with an AUC score reaching 0.80 to 0.83.

The boosting-gradient ensemble models have consistently outperformed LR, DT and RF. DT demonstrated significant improvement with an AUC Score between 0.67 and 0.74, while the LR model performed poorly with an AUC score of 0.59 to 0.60. The ROC curve, furthermore, demonstrated that SMOTE and ADSYN improve the performance of models across models. Overall, the results of the ROC curve show that boosting algorithms are well-suited for the real-world dataset because CatBoost can mitigate bias in categorical data and iteratively reduce noise from banking data.

The comparison of the six ML models combined with resampling methods and evaluation metrics demonstrated that the gradient-boosting ensemble methods are suitable for fraud detection under class imbalance. Gradient boosting methods demonstrate not only superior recall but also reliable balance between detecting fraudulent activity and controlling false positives, as indicated by their PR-AUC values in Table V. Among all evaluated methods, XGBoost with SMOTE outperformed the other five models, achieving a recall of 0.4667 and a precision of 0.1299, with the highest PR-AUC of 0.2224 across all configurations. This combination demonstrates that the model not only detects a meaningful proportion of fraudulent transactions but does so with a balance of precision that is operationally sustainable in the banking sector. Furthermore, the comparative experiment has demonstrated that resampling methods address critical challenges in fraud detection and are essential to achieve robust fraud detection performance in an imbalanced real-world dataset.

c) Results of the top five best-performing Models

Gradient boosting methods XGBoost with SMOTE was the best performing model with PR-AC of 0.2224. CatBoost with SMOTE was the second-best performing model with PR-AUC 0.2142, Precision: 0.1228 and Recall: 0.4395. LightGBM with SMOTE came third with PR-AUC: 0.2007, Precision: 0.1542 and Recall: 0.3778. XGBoost with SMOTE_TOMEK was in fourth position with PR-AUC:0.1971, Precision: 0.1175 and Recall:0.4321. While RF with no sampling strategy is the last best performing model with PR-AUC of 0.1923, Precision: 0.4037, and Recall: 0.2173. came second with a PR-AUC of 0.243, and RF, which was used as a baseline model, had a very low PR-AUC of 0.01822.

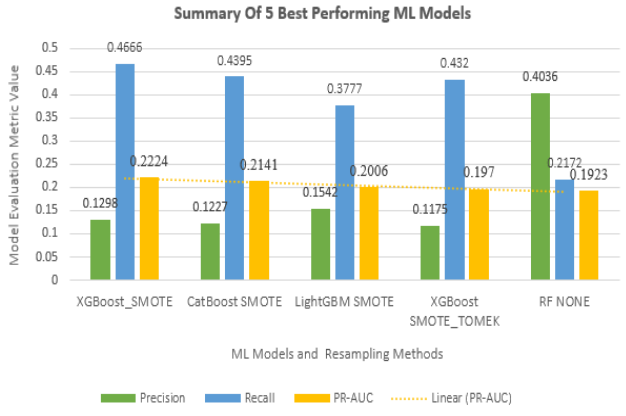


Fig. 3. Comparison of the five best-performing models

In terms of precision, once over-sampling methods were applied, precision values dropped into the 0.10 to 0.15 range, but recall substantially increased, making the outputs operationally more meaningful. The model was able to identify the intricate patterns of the minority class and make predictions that identify actual fraud. For instance, XGBoost with SMOTE produced a balanced outcome with a precision of 0.1299 and a recall of 0.4667. LightGBM with SMOTE exhibited similar behaviour, reaching a precision of 0.1542 and a recall of 0.3778, while CatBoost with SMOTE achieved a precision of 0.1227 and a recall of 0.4395. Overall, boosting ensembles like XGBoost and LightGBM excel in average precision because of their iterative ability to minimise errors. This preserves the majority class integrity better than bagging in Random Forest or Decision Tree.

The ability to detect actual fraud instances is essential in banking sector because it helps mitigate financial losses from undetected fraudulent transactions. In terms of recall, LightGBM yielded the highest score of 0.6988 with SMOTE-Tomek and 0.6123 with Borderline SMOTE. Recall values (Fig 4) demonstrated that the ensemble model outperformed baseline models and SMOTE boosted the model's ability to capture minority instances in the dataset.

The results demonstrated that the increase in recall was consistently associated with reductions in precision, indicating that the models had degenerated into predicting a large proportion of transactions as fraudulent, overwhelming practical utility. This setup would be ideal for the banking sector, where comprehensive fraud detection outweighs precision concerns. However, failure by baseline models emphasises the need to resample the training data and refine resampling methods.

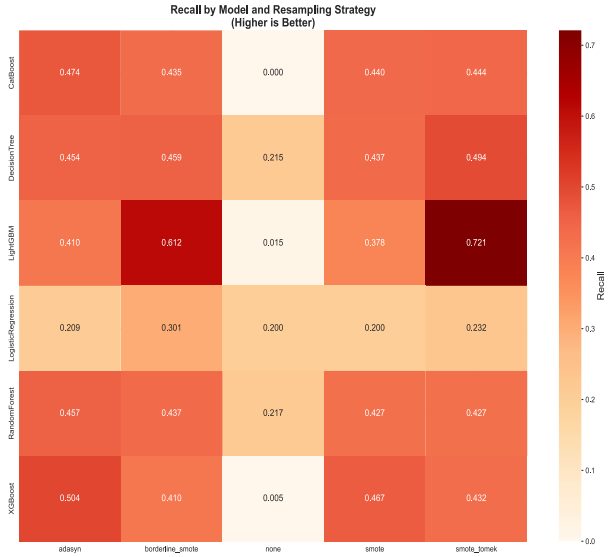


Fig. 4. Recall vs Resampling methods comparison of model performance

d) Comparative Analysis with Previous Studies

The comparative analysis of this study and related literature in Section III demonstrates the effectiveness of ensemble learning and resampling methods in fraud detection. However, the overall performance achieved in previous

studies was comparatively higher than the results obtained in this study. Kaushik et al. [28] presented an RF model achieving a PR-AUC of 0.855. The model was trained on a dataset with synthetically created features derived through Principal Component Analysis (PCA); no hyperparameter tuning was applied to the models. SMOTE was utilised to balance the data. The dataset used in this study was highly imbalanced, with 0.172% of records labelled as fraudulent.

Similarly, Saint [27] used a highly imbalanced synthetic dataset with 0.30 % fraudulent records. The study applied SMOTE to balance the data and feature engineering with error handling to enhance the model's predictive capabilities. RF model achieved PR-AUC of 09998. Detthamrong [26] used a synthetic dataset containing 0.2% fraudulent records. Hyperparameter tuning, sampling and scaling techniques were applied during model training. The approach resulted in XGBoost achieving the highest precision of 0.9756 and the second-highest recall of 0.8163.

In contrast, the best performing model in this study, XGBoost trained on data resampled using SMOTE, achieved a PR-AUC of 0.2222. While the ML models used in this study are consistent with those employed in previous studies, the results were different. Analysis of the results has demonstrated that the key factor contributing to the differences is the nature of the real-world dataset. Unlike the synthetic datasets used in previous studies, this study employed a highly imbalanced real-world dataset in which financial analysts manually identified and labelled fraudulent transactions. The dataset contained 114,746 observations; only 1.8 % were confirmed as fraudulent transactions. The possibility of unidentified fraudulent transactions is high. The exact proportion of the fraudulent transactions that were not identified is unknown. While synthetic datasets used in previous studies all have legitimate and fraudulent transactions known with certainty, which is different from real-world datasets, such as the one utilised in this study.

As discussed in this section, this study adopted a comprehensive methodological approach to ML modelling and predictive performance comparison. Unlike the previous studies reviewed in the related work section, which employed a single resampling method, this study used multiple resampling strategies alongside diverse ML models. In addition, this study employed domain expertise knowledge in feature engineering to extract domain-specific features to enhance predictive performance.

VIII. Conclusion

This study evaluated the performance of six ML models for fraud detection using a real-world dataset from the Malawian banking sector. By applying diverse algorithms and resampling methods, the study addressed class imbalance and achieved improved ML model predictive performance. The dataset captured transaction patterns characteristic of the African region, while domain-specific features were engineered using financial crime risk indicators.

Among the ML models employed in this study, XGBoost demonstrated consistent performance across F1 Score, Precision, and Accuracy metrics. However, the model achieved a lower recall value when trained on the SMOTE resampled dataset, compared with LightGBM, CatBoost, LR, DT and RF models. The results of the study highlight the effect of resampling methods on ML model performance and the transformative role of ML in fraud detection. This study provides a foundation for informed decision-making and policy formulation regarding applied ML for fraud detection in the banking sector.

The next study will extend beyond binary classification to advanced methods, such as autoencoders, to investigate the application of ML models in predicting anomalies and rare fraud patterns and improving model generalizability across different types of fraud.

Acknowledgment

We would like to thank everyone who helped write this paper and the Telkom Center of Excellence at the Nelson Mandela University for funding this research.

References

[1] D. O. Njoku, V. C. Iwuchukwu, J. E. Jibiri, C. T. Ikwuazom, C. I. Ofoegbu, and F. O. Nwokoma, "Machine learning approach for fraud detection system in financial institution: *A web base application," Machine Learning*, vol. 20, no. 4, pp. 01–12, 2024.

[2] F. Nujumudeen, V. Anuja, N. R. Devi, N. U. Madhuri, and N. K. Radha, "AI in Financial Services: Fraud Detection and Risk Management," *Metallurgical and Materials Engineering*, vol. 31, no. 3, pp. 119–127, Mar. 2025, doi: 10.63278/1335.

[3] J. Hu, Y. Zhang, and H. Zhang, "Hybrid optimisation and deep learning for enhancing accuracy in fraud detection using big data techniques," *Peer-to-Peer Networking and Applications*, vol. 18, no. 4, May 2025, doi: 10.1007/s12083-025-01971-4.

[4] A. Alsulami and R. Alabdan, "FRAUD DETECTION IN FINANCIAL TRANSACTIONS*," Advances and Applications in Statistics*, vol. 91, no. 8, pp. 969–986, May 2024, doi: 10.17654/0972361724052.

[5] S. Makki *et al.*, "An experimental study with imbalanced classification approaches for credit card fraud detection," *IEEE Access*, vol. 7, pp. 93010–93022, 2019. doi:10.1109/access.2019.2927266.

[6] A. A. Arfeen and B. M. Khan, "Empirical analysis of machine learning algorithms on detection of Fraudulent Electronic Fund transfer transactions," *IETE Journal of Research*, vol. 69, no. 11, pp. 7920–7932, Mar. 2022. doi:10.1080/03772063.2022.2048700

[7] P. Mrozek, J. Panneerselvam, and O. Bagdasar, "Efficient resampling for fraud detection during anonymised credit card transactions with unbalanced datasets," *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, pp. 426–433, Dec. 2020. doi:10.1109/ucc48980.2020.00067

[8] G. Y. Wong, F. H. F. Leung, and Sai-Ho Ling, "A novel evolutionary pre-processing method based on over-sampling and under-sampling for imbalanced datasets," *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, pp. 2354–2359, Nov. 2013. doi:10.1109/iecon.2013.6699499

[9] H. Ren, Y. Tang, W. Dong, S. Ren, and L. Jiang, *Duen: Dynamic ensemble handling class imbalance in network intrusion detection*, 2023. doi:10.2139/ssrn.4341598

[10] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Information Sciences*, vol. 513, pp. 429–441, Mar. 2020. doi:10.1016/j.ins.2019.11.004

[11] M. Ghanem *et al.*, "Limitations in evaluating machine learning models for imbalanced binary outcome classification in spine surgery: A systematic review," *Brain Sciences*, vol. 13, no. 12, p. 1723, Dec. 2023. doi:10.3390/brainsci13121723

[12] B. Baesens, S. Höppner, I. Ortner, and T. Verdonck, "Robrose: A robust approach for dealing with imbalanced data in fraud detection,"

Statistical Methods & Applications, vol. 30, no. 3, pp. 841–861, Jun. 2021. doi:10.1007/s10260-021-00573-7

[13] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, "Boosting methods for multi-class Imbalanced Data Classification: An experimental review," *Journal of Big Data*, vol. 7, no. 1, Sep. 2020. doi:10.1186/s40537-020-00349-y

[14] H. Wasswa, T. Lynar, and H. Abbass, "Enhancing IoT-botnet detection using variational auto-encoder and cost-sensitive learning: A deep learning approach for imbalanced datasets," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, Sep. 2023, pp. 1–6.

[15] The Registrar of Financial Institutions, "Financial Institutions Annual Report for 2024 ," Reserve Bank of Malawi, Aug. 2024. Accessed: 31 August, 2025. [Online]. Available: https://www.rbm.mw/Publications/AnnualReports/

[16] Reserve Bank of Malawi, "2024 National Payment System (NPS) Report," Reserve Bank of Malawi, Apr. 2025. Accessed: 31 August, 2025. [Online]. Available: https://www.rbm.mw/PaymentSystems/

[17] Financial Intelligence Authority (FIA) Malawi, "MONEY LAUNDERING TRENDS AND TYPOLOGIES REPORT JULY 2020 - MARCH 2022 ," Mar. 2025. Accessed: 31 August, 2025. [Online]. Available: https://www.fia.gov.mw/index.php

[18] S. Bhowmik and J. Howlader, "Monitoring and detection of NEFT fraudulent requests: A Comparative Analysis of Machine Learning Models," *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–8, Jun. 2024. doi:10.1109/icccnt61001.2024.10723320

[19] S. K. Hashemi, S. L. Mirtaheri, and S. Greco, "Fraud detection in banking data by Machine Learning Techniques," *IEEE Access*, vol. 11, pp. 3034–3043, 2023. doi:10.1109/access.2022.3232287

[20] Md. A. Talukder, M. Khalid, and M. A. Uddin, "An integrated multistage ensemble machine learning model for fraudulent transaction detection," *Journal of Big Data*, vol. 11, no. 1, Nov. 2024. doi:10.1186/s40537-024-00996-5

[21] A. A. Alhashmi, A. M. Alashjaee, A. A. Darem, A. F. Alanazi, and R. Effghi, "An ensemble-based fraud detection model for financial transaction Cyber Threat Classification and countermeasures," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12433–12439, Dec. 2023. doi:10.48084/etasr.6401

[22] M. Altalhan, A. Algarni, and M. Turki-Hadj Alouane, "Imbalanced Data Problem in Machine Learning: A Review," *IEEE Access*, vol. 13, pp. 13686–13699, 2025. doi:10.1109/access.2025.3531662

[23] M. Carvalho, A. J. Pinho, and S. Brás, "Resampling approaches to handle class imbalance: A review from a Data Perspective," *Journal of Big Data*, vol. 12, no. 1, Mar. 2025. doi:10.1186/s40537-025-01119-4

[24] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, Mar. 2019. doi:10.1186/s40537-019-0192-5

[25] M. Isangediok and K. Gajamannage, "Fraud Detection Using Optimised Machine Learning Tools Under Imbalance Classes," pp. 4275–4284, Sep. 2022, doi: 10.1109/BigData55660.2022.10020723.

[26] U. Detthamrong, W. Chansanam, T. Boongoen, and N. Iam-On, "Enhancing fraud detection in banking using Advanced Machine Learning Techniques," *International Journal of Economics and Financial Issues*, vol. 14, no. 5, pp. 177–184, Sep. 2024. doi:10.32479/ijefi.16613.

[27] A. F. Sariat, I. J. Siddique, M. Hossain, M. M. Islam, and T. Rahman, "AI driven fraud detection in financial ecosystems: A hybrid machine learning framework," *2025 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pp. 1–8, Feb. 2025. doi:10.1109/ecce64574.2025.11013808

[28] M. M. Kaushik, S. M. Mahmud, M. A. Kabir, and D. Nandi, "The effects of class rebalancing techniques on ensemble classifiers on credit card fraud detection: An empirical study," *AIP Conference Proceedings*, vol. 2916, p. 030011, 2023. doi:10.1063/5.0177524

[29] S. Ghosh, S. Bilgaiyan, M. K. Gourisaria, and A. Sharma, "Comparative analysis of applications of machine learning in credit card fraud detection," *2023 6th International Conference on Information Systems and Computer Networks (ISCON)*, pp. 1–7, Mar. 2023. doi:10.1109/iscon57294.2023.10112099

[30] M. F. Manzoor and A. Muhammad Faran, "Enhancing banking fraud detection: Role of machine learning and Deep Learning Methods," *Premier Journal of Artificial Intelligence*, 2025. doi:10.70389/pjai.100014

[31] M. E. Lokanan, "Predicting mobile money transaction fraud using machine learning algorithms," *Applied AI Letters*, vol. 4, no. 2, Apr. 2023. doi:10.1002/ail2.85

[32] S. Abdumalikov, J. Kim, and Y. Yoon, "Performance analysis and improvement of Machine Learning with various feature selection methods for EEG-based emotion classification," *Applied Sciences*, vol. 14, no. 22, p. 10511, Nov. 2024. doi:10.3390/app142210511

[33] R. Sibindi, R. W. Mwangi, and A. G. Waititu, "A boosting ensemble learning based hybrid light gradient boosting machine and extreme gradient boosting model for predicting house prices," *Engineering Reports*, vol. 5, no. 4, Nov. 2022. doi:10.1002/eng2.12599

[34] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 1937–1967, Aug. 2020. doi:10.1007/s10462-020-09896-5

[35] S. H. Sim, T. Paranjpe, N. Roberts, and M. Zhao, "Exploring Edge Machine Learning-based Stress Prediction using Wearable Devices," *Proceedings - 21st IEEE International Conference on Machine Learning and Applications, ICMLA 2022*, pp. 1266‑1273, 2022, doi: 10.1109/ICMLA55696.2022.00203.

[36] Y. Deng and T. Lumley, "Multiple imputation through xgboost," *Journal of Computational and Graphical Statistics*, vol. 33, no. 2, pp. 352–363, Oct. 2023. doi:10.1080/10618600.2023.2252501

[37] P. Zhang, Y. Jia, and Y. Shang, "Research and application of XGBoost in Imbalanced Data," International Journal of Distributed Sensor Networks, vol. 18, no. 6, p. 155013292211069, Jun. 2022. doi:10.1177/15501329221106935

[38] Y. Li, "Research on big data financial fraud detection system based on machine learning," *2024 5th International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, pp. 366–370, Sep. 2024. doi:10.1109/ichci63580.2024.10808078

[39] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.," The Annals of Statistics, vol. 29, no. 5, Oct. 2001. doi:10.1214/aos/1013203451

[40] L. Theodorakopoulos, A. Theodoropoulou, A. Tsimakis, and C. Halkiopoulos, "Big data-driven distributed machine learning for scalable credit card fraud detection using PySpark, XGBoost, and CatBoost," *Electronics*, vol. 14, no. 9, p. 1754, Apr. 2025. doi:10.3390/electronics14091754

[41] N. Jetty, P. V, S. Pattanaik and M. Aich, "Enhancing Fraud Detection in Financial Transactions: A Bio-Inspired Fine-Tuned CatBoost Approach," *2025 International Conference on Intelligent and Cloud Computing (ICoICC)*, Bhubaneswar, India, 2025, pp. 1-6, doi: 10.1109/ICoICC64033.2025.11052066

[42] A. Agarwal, S. Prabha, and R. Yadav, "Exploratory Data Analysis (EDA) for banking and finance: Unveiling insights and patterns," *GBS Impact: Journal of Multi Disciplinary Research*, vol. 11, no. 1, pp. 249–280, Jul. 2025. doi:10.58419/gbs.v11i1.1112520

[43] I. Gómez-Talal, M. Azizsoltani, P. Talón-Ballestero and A. Singh, "Machine Learning in Hospitality: Interpretable Forecasting of Booking Cancellations," in *IEEE Access*, vol. 13, pp. 26622-26638, 2025, doi: 10.1109/ACCESS.2025.3536094.

[44] D. Elreedy, A. F. Atiya, and F. Kamalov, "A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning," Mach. Learn., vol. 113, no. 7, pp. 4903–4923, Jul. 2024.

[45] A. Nugroho and H. Suhartanto, "Hyper-parameter tuning based on random search for DenseNet optimisation," *2020 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pp. 96–99, Sep. 2020. doi:10.1109/icitacee50144.2020.923916.

[46] H. Wen and F. Huang, "Personal Loan Fraud Detection Based on Hybrid Supervised and Unsupervised Learning," *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*, Xiamen, China, 2020, pp. 339-343, doi: 10.1109/ICBDA49040.2020.9101277.

Deep Learning Driven Threat Detection in Network Traffic Using Multi-Layer Perceptron (MLP) on Public Intrusion Datasets

Tinashe Crispen Garidzira
Department of Computational Sciences
University of Fort Hare
Alice, South Africa
0009-0003-4850-1568

Courage Matobobo
Department of Business and Application Development
Walter Sisulu University
East London, South Africa
0000-0002-7125-5989

William Tichaona Vambe
Department of Mathematical Sciences and Computing
Walter Sisulu University and The National Institute for Theoretical and Computational Sciences
Mthatha, South Africa
0000-0003-0516-1260

Caroline Gurajena
Department of Computational Sciences
University of Fort Hare
Alice, South Africa
cgurajena@ufh.ac.za

***Abstract*— With the proliferation of the internet and the widespread use of cloud services, as well as the increasing reliance on online platforms for various activities, there has been a corresponding rise in the sophistication and frequency of cyber threats. This means intrusion detection will remain an important component in safeguarding digital assets and network infrastructures. Intrinsically, this study introduces a data-driven Deep Learning approach for threat detection in network traffic using a Multi-Layer Perceptron architecture guided by the Cross-Industry Standard Process for Data Mining methodology. Publicly available CIC-IDS2017 intrusion detection dataset from Kaggle was used, and the raw traffic data was preprocessed and analyzed for threat classification. To enhance model efficiency and reduce dimensionality, the Extreme Gradient Boosting algorithm was used to select relevant features by reducing the original 78 features to the 9 most significant attributes based on importance ranking. The selected features were normalized and used to train and evaluate the Multi-Layer Perceptron model. Experimental results demonstrate that the proposed model achieved a test accuracy of 96.36% with a corresponding loss of 0.135, highlighting its strong capability in distinguishing between benign and malicious traffic flows. This work underscores the potential of combining feature selection with lightweight DL models for developing scalable, accurate, and cost-effective intrusion detection systems in modern network environments.**

Keywords*— *Intrusion Detection System (IDS), Deep Learning (DL), Multi-Layer Perceptron (MLP), Network Security, CIC-IDS2017 Dataset

I. INTRODUCTION

Network security has become increasingly critical in recent years due to the proliferation of the internet and the use of cloud services. Many people are now using the internet for almost everything, which exposes users to sophisticated cyber threats targeting digital assets and digital infrastructures. As such Intrusion Detection Systems (IDS) becomes key. Intrusion Detection Systems (IDS) are important for maintaining the integrity, confidentiality, and availability of modern network infrastructures without cyber threats. Intrusion Detection Systems serve as a key defence mechanism, by identifying unauthorized activities and mitigating attacks before significant harm occurs [1], [2]. Traditional signature-based IDS methods have been used over the years. However, they struggle to detect sophisticated, novel, or adaptive attacks commonly observed in dynamic network environments [2]. To address the weaknesses of traditional IDS methods, advanced data-driven approaches, such as Machine Learning (ML) and Deep Learning (DL) have been adopted. Notably, there has been a significant improvement in the performance of IDS, enabling them to identify complex and evolving attack patterns more effectively [2], [3].

Deep Learning (DL) models can automatically learn hierarchical feature representations from high-dimensional data, such as network traffic. These models outperform conventional shallow models in both detection accuracy and adaptability [1], [2]. Among these DL techniques, the Multi-Layer Perceptron (MLP), a type of feed-forward neural network, has gained renewed attention given its balance of simplicity and performance. Although MLP architectures are simple and lightweight, they are effective in intrusion detection tasks [4]. Studies such as [4], [5], [6] have utilized MLP-based architectures on the CIC-IDS2017 dataset, and have reported impressive intrusion detection rates, with some achieving accuracy above 99% and low false positive rates.

Although DL techniques have strengths in the high dimensionality and redundancy present in intrusion datasets like CIC-IDS2017, they suffer from learning efficiency and increased computational costs [7]. As such, there is a need to address these computational efficiency problems by using feature selection methods. This is very important to reduce the dimensionality of the data by removing some of the features that add noise. Tree-based techniques like Extreme Gradient Boosting (XGBoost), have shown considerable efficacy in feature importance separation [8]. These algorithms not only improve classification performance but also significantly reduce dimensionality [8], [9]. Moreover, hybrid frameworks that combine methods like SMOTE for class balancing and XGBoost for feature selection have achieved near-perfect accuracy and robust generalization on benchmark datasets [8].

A study by [6], proposed a two-stage methodology to enhance MLP-based intrusion detection using the CIC-IDS2017 dataset. They applied unsupervised clustering using Birch or K-means to the dataset to generate pseudo-labels, which were then incorporated as additional features during MLP model training. Their experimental results indicated an improvement in multi-class classification accuracy by achieving an accuracy of 99.73%, outperforming standalone MLP models. The study leveraged clustering to enrich feature representation and boost detection performance without greatly increasing complexity. However, the limitation of

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

their work was the reliance on clustering-generated pseudo-labels, which may introduce bias or overfitting, and generalizability on other datasets or real-world network traffic remains untested.

Researchers [10] proposed a hybrid feature selection method information gain and Random Forest Recursively removing features (IGRF-RFE) made for MLP-based intrusion detection using the UNSW-NB15 dataset. They combined filter techniques, Information Gain (IG), and Random Forest (RF), to reduce the initial feature space, followed by Recursive Feature Elimination (RFE) as a wrapper to further refine the feature subset. This process reduced the number of features from 42 to 23, and improved multi-class classification metrics, leading to MLP increasing accuracy from 82.25% to 84.24%. One of the key strengths of their approach was that it used a dual-stage selection that balances speed with relevance, obtaining incremental performance gains with dimensionality reduced. However, the improvement was relatively modest, and their approach only tested UNSW-NB15, not on CIC-IDS2017, limiting generalizability across other datasets.

A study in [2] conducted a controlled comparison of several models, including MLP, 1D CNN, One-Class SVM (OCSVM), and Local Outlier Factor (LOF) on the CIC-IDS2017 dataset. They evaluated both the detection of known attack types and generalization to unseen threats. The experimental results indicated that supervised models (MLP and CNN) achieved good accuracy on familiar attacks. However, they struggle with drastic recall drops on novel attack types. On the other hand, LOF displayed moderate overall accuracy of 60.47% and low recall of 57.46%on unknown threats, while OCSVM offered a better accuracy and F-1 score of 79.19% and 75.75%, respectively. A major strength of their study was the dual-scenario evaluation, showing the trade-offs between accuracy and generalization across different contexts. However, they did not pair MLP with any feature selection technique, thus, missing optimization in efficiency or dimensionality reduction.

While prior studies have explored MLP architectures for intrusion detection, our motivation is driven not by architectural novelty but the pursuit of enhanced computational efficiency and optimized feature representation. For instance, the study in [6] used MLP detection performance using clustering, but their reliance on pseudo-labelling raises concerns about overfitting and generalizability. Similarly, study [10] introduced a hybrid feature selection method to improve MLP accuracy, yet their results were modest and restricted to UNSW-NB15, limiting transferability to other datasets. On the other hand, [2] did a comparison between MLP and CNN on anomaly detection methods, demonstrating useful insights into model trade-offs, but without applying feature selection to optimize MLP's efficiency. Therefore, this work seeks to apply XGBoost-based feature selection to reduce redundancy in CIC-IDS2017 and identify the most significant attributes for intrusion detection. This study is guided by the following key objectives:

1. Develop and train an optimized MLP model on the reduced feature set to achieve high detection accuracy with lower computational cost.
2. Test and evaluate the model, accuracy rate, efficiency, and reliability of the model in detecting.

The remainder of this paper is structured as follows: Section II presents related works, followed by the methodology adopted, Section III. Results and discussions are presented in Section IV. Section V concludes the paper and presents future research directions.

II. Related Works

Several works have been done using DL for threat detection. Researchers in [11] introduced a novel Explainable AI (XAI)-based feature selection method made for IDS. They applied several XAI techniques to identify and rank the most important features across standard datasets like CIC-IDS2017 and then compared their method against conventional feature selection strategies. Their proposed XAI-driven approach consistently improved model performance across diverse classifiers. Moreover, enhanced the interpretability of the selected features, enabling security analysts to better understand and justify IDS decisions. However, they did not integrate the feature selection into a specific lightweight DL architecture like MLP, leaving a gap for future integration into efficient models.

The works of [12] developed an intrusion detection framework (IDF) for Internet of Things (IoT). The researchers used XGBoost on the CIC-IDS2017 and CICIoT2023 datasets. Furthermore, they integrated model explainability via rule induction to create a transparent decision-making process. The experimental results showed that XGBoost achieved 99.91% accuracy and 99.88% AUC-ROC on the CIC-IDS2017 dataset, as well as 98.54% accuracy and 93.06% AUC-ROC on CICIoT2023 dataset, One of the major strengths of their study was combining high detection performance with interpretability, making it more trustworthy and applicable in operational environments. On the other hand, their proposed framework did not involve MLP-based models, and they focused only on XGBoost, limiting direct comparison to the MLP approach.

Another study on ML models for network intrusion detection by [8], developed a hybrid framework for intrusion detection by combining SMOTE for class imbalance, XGBoost for feature selection, and hybrid ML/DL classification on datasets like KDD-CUP'99 and CIC-MalMem-2022. Their experimental results indicated that the method they used achieved an accuracy of 99.99% on KDD-CUP'99 and 100% on CIC-MalMem-2022, with no evidence of overfitting. A major strength of their study was the pipeline they used, which addressed imbalance and feature reduction alongside high detection performance. However, their study used a different dataset from CIC-IDS2017, and no MLP-specific architecture was evaluated, limiting direct comparability.

Researchers [13] proposed an efficient intrusion detection model, which was based on hybrid feature selection and stack ensemble learning. They named their model MI-Boruta and it combined mutual information (MI) as a filter method and the Boruta algorithm as a wrapper method to determine optimal features from our datasets. Their approach showed better performance across multiple metrics with 95.34% accuracy rate, 94.64% recall, 92.62% precision, F1 score of 93.62%, and low false positive rates, outperforming single-algorithm IDS models. The strength of their study was the

integration of robust feature selection and ensemble layering with MLP, improving generalizability across datasets. On the other hand, they increased architectural complexity and potentially reducing suitability for lightweight scenarios.

In a study by [14] on IDS feature selection and ensemble classifier, the researchers introduced an IDF combining Correlation Feature Selection through Bat Algorithm (CFS-BA) for dimensionality reduction, followed by an ensemble classifier using C4.5, Random Forest, and Forest PA, to detect intrusions. They tested the framework on NSL-KDD, AWID, and CIC-IDS2017 datasets. The method outperformed several DL approaches across different classification metrics with an accuracy of 99.8%. One of the major strength of their study was the use of optimized feature selection and ensemble consensus to enhance detection accuracy on CIC-IDS2017. However, the study did explore MLP models, limiting insights into modern DL integration.

A comprehensive review by [15] examined developments in DL applications for intrusion detection, focusing on challenges of handling complex spatiotemporal features and addressing data imbalance issues across multiple standard datasets, including CIC-IDS2017. The review showed that strategies like attention-based architectures, data augmentation, and class imbalance handling has been used but indicated that there still exist gaps in current MLP models for IDS. The work synthesized key challenges and guiding design principles for future DL-based IDS research. One of the major limitation of their study was lack of empirical evaluation of feature selection or specific model pipelines like MLP and XGBoost approach.

From existing literature, an IDF that combines lightweight DL architectures such as MLP with feature selection (e.g., XGBoost or Boruta) on CIC-IDS2017 has not yet been fully realized. Previous studies [8], [11], [12], [13], [14]among others, have focused on feature selection and ensemble-based approaches to improve detection accuracy and interpretability. These works demonstrate the possibility of combining advanced feature engineering with traditional ML or ensemble learners. However, their deployment strategies remain constrained either to non-MLP classifiers or to complex ensemble pipelines that may reduce efficiency. For example, [15] highlighted the challenges of spatiotemporal features and class imbalance in DL-based IDS but did not use lightweight implementations.

Therefore, this study seeks to bridge this gap by proposing a DL-driven IDS using MLP with optimized feature selection, ensuring both high detection performance and computational efficiency for practical network security applications.

III. METHODOLOGY

This study adopted the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology and was guided by the six stages namely: business (problem) understanding, data understanding, data preparation, model building, evaluation results, and deployment [16]. Cross-Industry Standard Process for Data Mining was best for this work because it offers "enhanced data understanding and preparation" which can be tailored to network traffic, and better model evaluation before deployment, leading to more accurate and effective DL-based threat detection as articulated by [10].

The publicly available dataset titled "Intrusion detection evaluation dataset (CIC-IDS2017)" [17] was used for this study. The dataset contained csv files, , where each csv file contains data of network activities that were happening on that specific day. Instead of using the entire dataset, a file named *Wednesday-workingHours.pcap_ISCX.csv* was selected, which contains network attacks and normal activities that were happening on Wednesday during working hours.

The original dataset file used in this study was named *Wednesday-workingHours.pcap_ISCX.csv*, which contained 79 columns, consisting of 78 feature columns and a single label column. Several rows of the dataset contained missing (NA) and infinite values, which were removed as a preprocessing step to make sure that they will not have missing or unrealistic values such as infinity, as supported by [18]. All feature columns were numeric, except for the label column, which was categorical. The dataset was then split into 2 subsets, which are training and testing, with 30% of the data reserved for testing, following common practices in ML research as alluded to in [19].

Features and labels were separated, and the categorical labels were encoded into numerical form using the LabelEncoder from the scikit-learn library. This was done because ML and DL models require numerical inputs, as noted also in [15]. The seventy eight (78) numeric features were standardized using the StandardScaler, applied through a ColumnTransformer, to ensure they were normalized based on mean and standard deviation, thereby improving model convergence. Both the training and testing feature sets were scaled consistently to maintain distributional integrity across sets. The processed features were used to train an XGBoost-based feature selector, which identified the most significant attributes contributing to intrusion detection, leveraging its ability to rank features by importance as noted in [12]. The XGBoost algorithm was used to rank all 78 features based on their contribution to classification. A feature importance threshold of 2% was adopted, retaining only attributes with importance values equal to or above this cutoff. Table I shows the feature importance that was extracted using XGBoost.

TABLE I. IMPORTANCE OF EACH FEATURE IN DETECTING INTRUSIONS

Feature	Importance
BwdPackets/s	0.397280
BwdPacketLengthStd	0.096771
MinPacketLength	0.077367
DestinationPort	0.069533
PSHFlagCount	0.057434
FwdIATTotal	0.032330
FINFlagCount	0.024655
FwdPSHFlags	0.024299
act_data_pkt_fwd	0.023209

Out of the original 78 features, nine (9) exceeded the 2% importance threshold that was set as shown in Table I. The most significant feature was *BwdPacket/s*, with an important score of 39.73 %, followed by *BwdPacketLengthStd* at 9.68 %, while *act_data_pkt_fwd* had the lowest among selected features at 2.32 %. Fig. 1 visually represents these feature rankings inform of a bar chart.

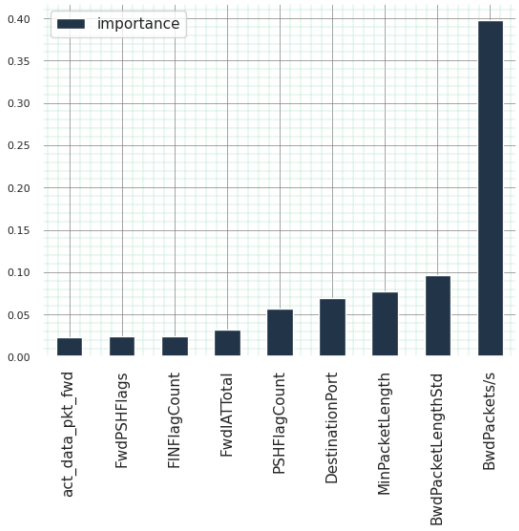


Fig. 1. Thread detection feature importance

Fig. 1 shows the trend increase of how each feature among the selected 9 was in detecting threats in network traffic. The rankings were increasing starting from *act_data_pkt_fwd*, *FwdPSHFlags*, *FINFlagCount*, *FwdIATTotal*, *PSHFlagCount*, *DestinationPort*, *MinPacketLength*, *BwdPacketLengthStd*, all the way to *BwdPackets/s* as shown in Fig. 1. After feature selection, visualization was done to the distribution of the target labels to visually see class balance. Fig. 2 shows the distribution of these target attack labels using a pie chart.

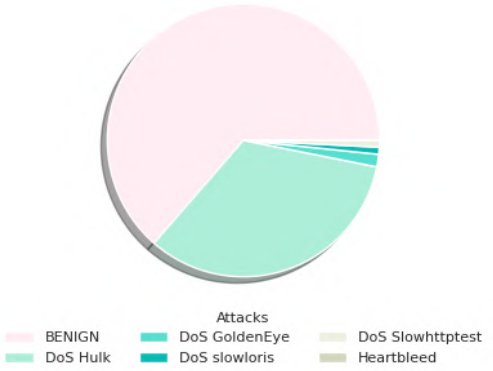


Fig. 2. Distribution of threats in the original dataset.

As shown in Fig. 2, the dataset displayed a visible class imbalance with the BENIGN class being the most dominant while Heartbleed represented a syndrome with very few samples. To address this, the Heartbleed class was excluded due to its insufficient representation, enabling the rebalancing of the remaining classes. This adjustment allowed the model to focus on learning to distinguish DoS attack types such as *GoldenEye*, *Slowhttptest*, *Hulk*, and *Slowloris* from the BENIGN class. Addressing class imbalance in IDS datasets is known to significantly enhance detection performance, particularly in DL contexts, as supported by [20]. Methods like undersampling, oversampling, and their hybrids are commonly employed. To address this imbalance, a down-sampling method was applied to fit the class to a minority of examples, ensuring a balanced dataset [21]. Studies [20], [22] showed that such techniques improve metrics like F1-score and G-mean across IDS classifiers. Fig. 3 shows the distribution of labels in the balanced dataset.

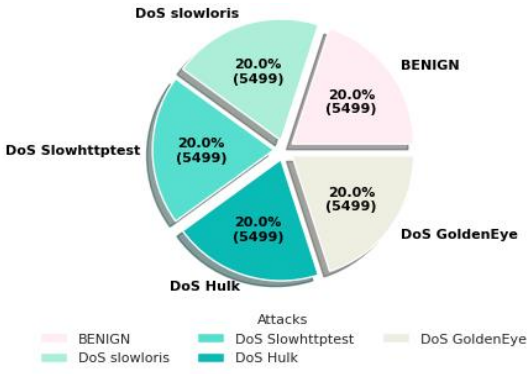


Fig. 3. Distribution of threats after balancing the dataset.

As displayed in Fig. 3, the dataset was balanced using the down-sampling approach, leading to each class BENIGN and four types of DoS attacks to have 20% of the samples summing to 5499 examples per class. Although each class was down sampled to 5499 samples before splitting, slight variations in the number of examples per class appear in the evaluation reports due to stratified splitting. After balancing the dataset based on the minority class, the dataset was split into training, validation, and test sets as shown in Fig. 4.

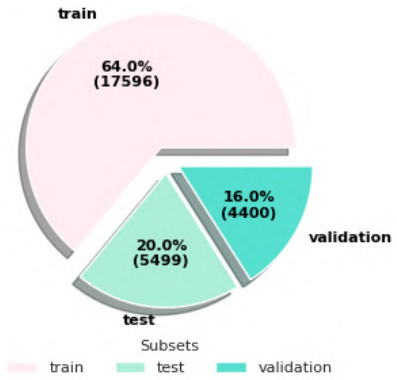


Fig. 4. Distribution of samples across sets.

As shown in Fig. 4, the split ratio of 80:20 for model development (training) and testing subset respectively was performed on the balanced dataset, and rounding during sample allocation across train, validation, and test sets caused minor differences in support counts. From the model development subset, 20% was taken and allocated to the validation set, which settled a fraction of 16% of the 64% for training. This 64, 16, 20 split balances the need for sufficient learning capacity with reliable model evaluation as supported in [23]. Employing a separate validation set allows for iterative hyperparameter tuning and model assessment without using the final test set as noted in [24], which remains untouched to preserve evaluation integrity. Such a structure is especially valuable in DL and intrusion detection scenarios to prevent overfitting and ensure generalizability [25]. Fig. 5 shows the distribution of labels in each subset split.

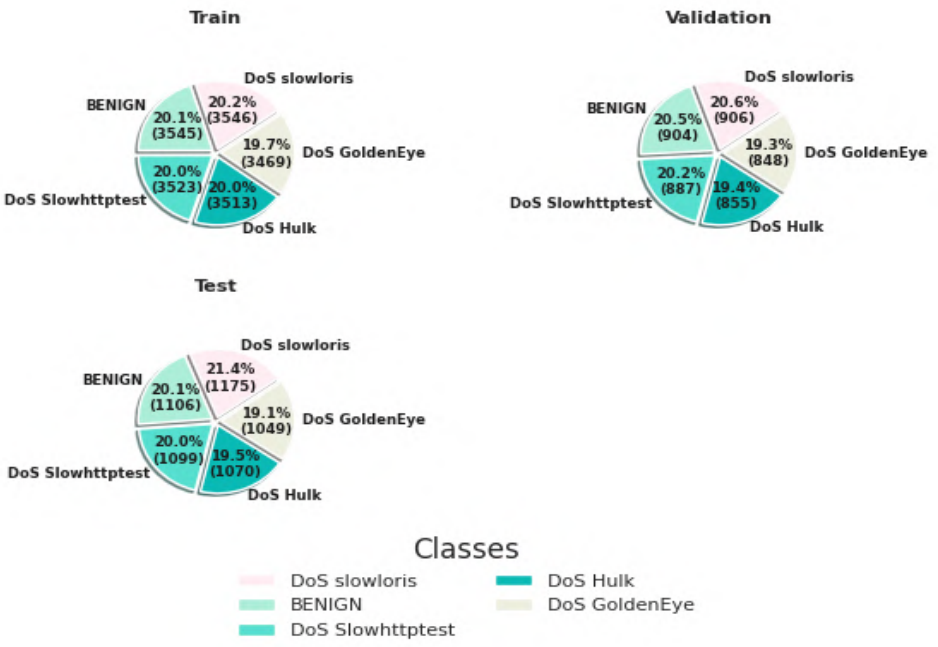


Fig. 5. Distribution of labels in each subset

Fig. 5 shows the uniformity in distribution of labels across all sets, training, testing, and validation. This shows that there will be no class imbalance during model training, validation, and final model evaluation. After the data preprocessing step, an MLP model was created using PyTorch subclassing method for flexibility. The MLP model took in nine (9) selected features that map them into a 128-dimensional latent space in the first Linear layer. A ReLU activation function introduces non-linearity, followed by a 50% Dropout layer to reduce overfitting and enhance generalization [26]. The resulting features were then reduced into a 16-dimensional representation, and passed through ReLU activation and 50% dropout. The final linear layer then projects 16-dimensional embedding into five (5) output classes, which corresponds to the selected target labels. Table II shows a complete summary of the MLP parameters.

TABLE II. MLP MODEL PARAMETERS

Parameters	Value
Total	3 429
Trainable	3 429

Our simple MLP model for network threat detection contained a total of 3429 parameters, and all parameters were trainable as shown in Table II. The MLP model was then trained for 40 epochs. The model performance was tracked for each training iteration for performance measurement, and the metrics that were tracked were epoch training time, training and validation losses, and accuracy. Model checkpointing was used to keep the best model that was observed during model training. The best-performing model was saved when validation loss reached its minimum, a common technique to preserve with the best weights and prevent overfitting as documented in [27]. The best saved MLP model was then evaluated on the testing dataset using metrics such as test loss, accuracy, a confusion matrix, and a classification report.

IV. RESULTS AND DISCUSSION

This section of the study presents experimental results. Metrics such as validation loss and accuracy trends are important measures of model generalization and reliability, as highlighted in other intrusion detection studies [28]. The overall training summary was visualised using a table, then epoch-wise plots showing loss, accuracy epoch-wise training time over 40 iterations. The checkpointing strategy was used to preserve the model weights with the lowest validation loss. This is a widely recognized method to mitigate overfitting and ensure optimal model selection [29]. The final saved model was then evaluated on the testing dataset using metrics, such as testing accuracy, testing loss, confusion matrix, and classification report. These metrics enabled us to evaluate the model performance in line with best practices in IDS evaluation as supported in [30]. Table III shows the MLP training summary.

TABLE III. MLP TOTAL TRAINING TIME AND LAST CHECKPOINTED EPOCH.

Metric	Value
Total Epochs	40
Last Saved Epoch	34
Total Training Time	1 minute 32.13 seconds

As shown in Table III, the MLP model completed training in exactly 1 minute and 32.13 seconds after training for 40 epochs on a standard Google Colab GPU. The model achieved a lowest validation loss on epoch 34, on which the best model was saved. This strategy of model checkpointing was used because it is known to optimize generalization [31], [32] by selecting the best model observed during training. Fig. 6 shows the visual representation of the model's epoch-wise training time.

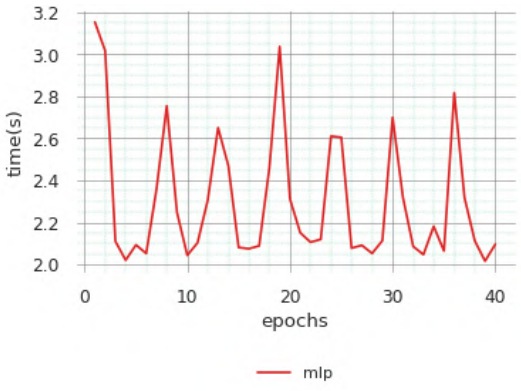


Fig. 6. Epoch training time

The line graph in Fig. 6 shows that the first epoch took more than 3.1 seconds, while other epochs were faster, completing training under 2.6 seconds on average. This initial delay in model completion on the first epoch aligns with expanded observations in DL training pipelines, where the first epoch is commonly slower due to data loading overheads and initialization . It is normal, as echoed in [33], that the first epoch takes more time to train because of caching mechanisms and dataset prefetching, once data resides in cache, and epochs that follow will train faster. Fig. 7 shows the MLP epoch-wise training loss.

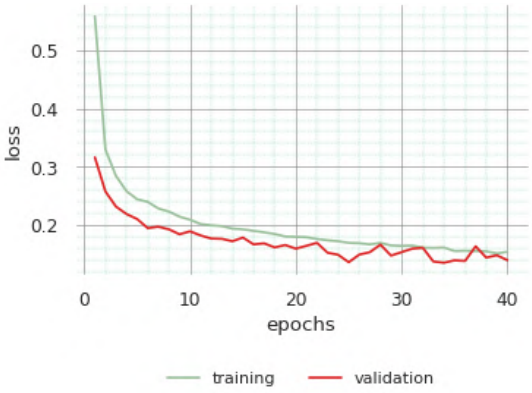


Fig. 7. MLP validation and training losses

As shown in Fig. 7, the validation loss began slightly above 0.3 while the training loss started higher, above 0.5, on the first epoch. Starting from the second epoch, both validation and training losses steadily decreased over the period of 40 epochs. The lowest validation loss was observed at epoch 34, indicating convergence and optimal generalization. The analysis of the trends in training and validation loss across epochs using line graphs provides important insight into model behaviours such as overfitting, underfitting, and stable learning. Fig. 8 shows the MLP training and validation accuracy per epoch for the MLP that was monitored during model training.



Fig. 8. MLP validation and training accuracies

It can be visually seen in Fig. 8 that the model's validation accuracy started at a higher value of above 90% while the training accuracy started at a value just above 82% but below 85%. Both training accuracy and validation accuracy steadily increased throughout 40 epochs. This indicates that the model effectively learned the relationship between input features and network attack labels. Validation accuracy exceeded training accuracy from the initial epoch to the last epoch, suggesting the absence of overfitting and the presence of generalization. Table IV shows the best saved MLP model evaluation metrics in terms of loss and accuracy based on the testing dataset.

TABLE IV. BEST SAVED MLP MODEL METRICS ON THE TESTING SET.

Metric	Value
Accuracy	96.36%
Loss	0.135

The model achieved a classification accuracy of 96.36 with a corresponding loss of 0.135. These results show that the model successfully learned to classify the selected features into their respective threat labels. This demonstrates effective feature label mapping and feature selection within the intrusion detection task. Table V shows the comparison. To contextualize the performance of the proposed MLP and XGBoost model used in this study, Table V summarizes comparison with similar studies that employed MLP or hybrid IDS frameworks.

TABLE V. COMPARATIVE WORKS THAT EMPLOYED MLP IN IDS

Work	Accuracy Score (%)
Current Study	96.36
Yin et al. [6]	99.73
Alsaffar et al. [13]	95.34

The proposed model achieved 96.36% accuracy, comparable to a study in [6] (99.73%) and [13] (95.34%) as shown in Table V while requiring fewer parameters and shorter training time. This indicates that our optimized MLP architecture achieves competitive performance at a lower computational cost, supporting its suitability for real time IDS applications. To provide a clearer understanding of the classification performance of the best saved MLP, Fig. 9 shows the confusion matrix generated on the testing dataset. The confusion matrix visually illustrates the distribution of correct and incorrect predictions across the classes.

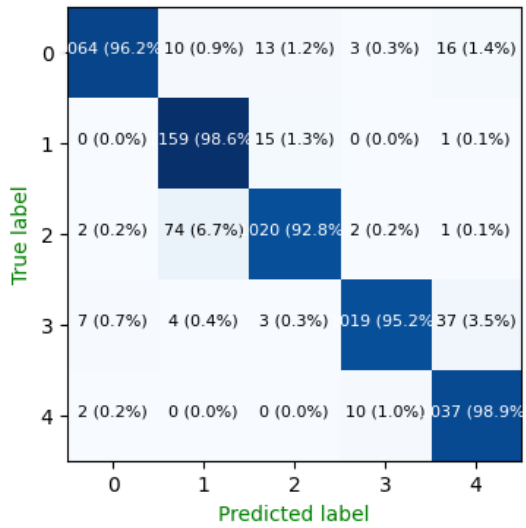


Fig. 9. Best saved MLP model Confusion Matrix on the testing set.

The confusion matrix shown in Fig. 9 demonstrates that most misclassifications occurred among DoS variants, indicating overlapping traffic signatures within these categories. Despite this, the high recall for BENIGN traffic (96.2%) and low false positive rate demonstrate the model's reliability for real-world deployment, where minimizing false alarms is critical. The model effectively distinguishes normal traffic from malicious flows, though future work should further address fine-grained discrimination among similar DoS attacks. Such confusion is common in IDS because similar DoS patterns often share overlapping traffic characteristics, making class separation difficult. Although overall performance is robust in minimizing false positives for benign traffic and improving class-specific recall for attack variants, the use of confusion matrices is important rather than relying solely on global accuracy [34]. To enhance readability, the classification metrics were reformatted into a classification report in Fig. 10 reports the precision, recall, and F1-score per class.

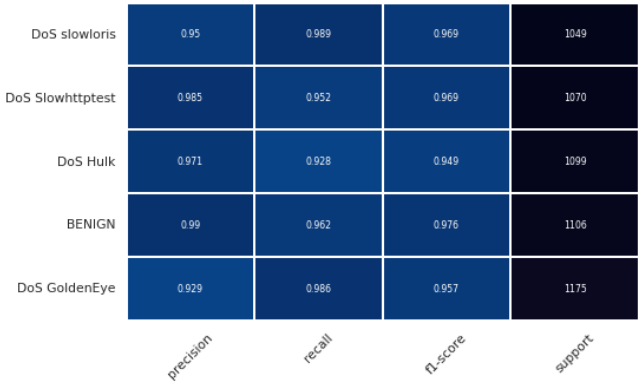


Fig. 10. Best saved MLP model Classification Report on the testing set.

The classification report in Fig. 10 shows that the model performs strongly across all classes. The *BENIGN* traffic achieved the highest F1-score of 0.976, while *DoS Slowloris* had an F1-score of 0.969 and *DoS Slowhttptest* had an F1-score of 0.969. *DoS GoldenEye* has a slightly lower precision of 0.929 but with a high recall of 0.986 whereas *DoS Hulk* had the lowest recall of 0.928. This suggests occasional misclassifications of *DoS Hulk*, though its F1-score remains 0.949. Overall, the best saved MLP model demonstrates good capability in distinguishing benign from malicious traffic. Although minor overlaps between DoS attack types remained a challenge, consistently highlighted in intrusion detection research [15], [34].

V. Conclusion and Future Works

This study presented a DL IDS using MLP combined with XGBoost-based feature selection on the CIC-IDS2017 dataset. By reducing the original dataset from 78 features to the 9 most important, the proposed model achieved a test accuracy of 96.36% with a loss of 0.135. This demonstrates the effectiveness of the proposed pipeline in classifying benign and malicious traffic threats. The confusion matrix and classification report confirmed strong detection rates across all classes, with *BENIGN* traffic achieving an F1-score of 0.976, while *DoS Slowloris* and *DoS Slowhttptest* both scored 0.969. Although there were a few misclassifications which occurred between similar DoS attacks, the MLP's overall performance remained robust. This reflects findings in IDS research where lightweight architectures such as MLP balance simplicity, computational efficiency and accuracy. The results showed the significance of feature selection in enhancing model generalization and computational efficiency. This aligned with other studies that emphasized combining feature engineering with scalable DL for intrusion detection.

Although the proposed model pipeline achieved good results, future research should investigate integrating explainable AI techniques to enhance the interpretability of MLP-based intrusion detection systems as suggested by. The future works should also take directions in experiments to additional datasets, such as UNSW-NB15 and CICIoT2023, that will also improve the generalizability of the approach. Moreover, exploring hybrid frameworks that combine clustering, ensemble learning, or attention-based architectures can further improve the detection of closely related DoS variants and unseen attack types. Another future direction of this research is deploying the model in real-time environments, particularly for IoT and edge computing scenarios. This is an important direction to evaluate scalability and resilience against evolving cyber threats.

VI. Acknowledgements

The authors would like to express their sincere gratitude to the creators and maintainers of the Canadian Institute for Cybersecurity (CIC) and the University of New Brunswick (UNB) for providing the CIC-IDS2017 dataset [17], which served as the foundation of this research. Their efforts in making high-quality benchmark datasets publicly available have been instrumental in advancing intrusion detection research. The authors also want to thank Telkom Centre of Excellence, University of Fort Hare, Walter Sisulu University and The National Institute for Theoretical (NITheCS) for the support.

VII. References

[1] R. Chinnasamy, M. Subramanian, S. V. Easwaramoorthy, and J. Cho, "Deep learning-driven methods for network-based intrusion detection systems: A systematic review," *ICT Express*, vol. 11, no. 1, pp. 181–215, Feb. 2025, doi: 10.1016/J.ICTE.2025.01.005.

[2] Z. Xu and Y. Liu, "Robust Anomaly Detection in Network Traffic: Evaluating Machine Learning Models on CICIDS2017," Aug. 2025, [Online]. Available: http://arxiv.org/abs/2506.19877

[3] H. M. Rai, J. Yoo, and S. Agarwal, "The Improved Network Intrusion Detection Techniques Using the Feature

Engineering Approach with Boosting Classifiers," *Mathematics*, vol. 12, no. 24, p. 3909, Dec. 2024, doi: 10.3390/math12243909.

[4] S. Cherfi, A. Lemouari, and A. Boulaiche, "MLP-Based Intrusion Detection for Securing IoT Networks," *Journal of Network and Systems Management*, vol. 33, no. 1, p. 20, Jan. 2025, doi: 10.1007/s10922-024-09889-7.

[5] A. Rosay, K. Riou, F. Carlier, and P. Leroux, "Multi-layer perceptron for network intrusion detection," *Annals of Telecommunications*, vol. 77, no. 5–6, pp. 371–394, Jun. 2022, doi: 10.1007/s12243-021-00852-0.

[6] Y. Yin, J. Jang-Jaccard, F. Sabrina, and J. Kwak, "Improving Multilayer-Perceptron(MLP)-based Network Anomaly Detection with Birch Clustering on CICIDS-2017 Dataset," Oct. 2022, [Online]. Available: http://arxiv.org/abs/2208.09711

[7] M. Deng, C. Sun, Y. Kan, H. Xu, X. Zhou, and S. Fan, "Network Intrusion Detection Based on Deep Belief Network Broad Equalization Learning System," *Electronics (Basel)*, vol. 13, no. 15, p. 3014, Jul. 2024, doi: 10.3390/electronics13153014.

[8] Md. A. Talukder *et al.*, "A Dependable Hybrid Machine Learning Model for Network Intrusion Detection," Jan. 2023, doi: 10.1016/j.jisa.2022.103405.

[9] E. Emirmahmutoğlu and Y. Atay, "A feature selection-driven machine learning framework for anomaly-based intrusion detection systems," *Peer Peer Netw Appl*, vol. 18, no. 3, p. 161, May 2025, doi: 10.1007/s12083-025-01947-4.

[10] Y. Yin *et al.*, "IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *J Big Data*, vol. 10, no. 1, p. 15, Feb. 2023, doi: 10.1186/s40537-023-00694-8.

[11] O. Arreche, T. Guntur, and M. Abdallah, "XAI-based Feature Selection for Improved Network Intrusion Detection Systems," Oct. 2024, [Online]. Available: http://arxiv.org/abs/2410.10050

[12] K. S. Adewole, A. Jacobsson, and P. Davidsson, "Intrusion Detection Framework for Internet of Things with Rule Induction for Model Explanation," *Sensors*, vol. 25, no. 6, p. 1845, Mar. 2025, doi: 10.3390/s25061845.

[13] A. M. Alsaffar, M. Nouri-Baygi, and H. M. Zolbanin, "Shielding networks: enhancing intrusion detection with hybrid feature selection and stack ensemble learning," *J Big Data*, vol. 11, no. 1, p. 133, Sep. 2024, doi: 10.1186/s40537-024-00994-7.

[14] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier," Apr. 2020, doi: 10.1016/j.comnet.2020.107247.

[15] Y. Zhang, R. C. Muniyandi, and F. Qamar, "A Review of Deep Learning Applications in Intrusion Detection Systems: Overcoming Challenges in Spatiotemporal Feature Extraction and Data Imbalance," *Applied Sciences*, vol. 15, no. 3, p. 1552, Feb. 2025, doi: 10.3390/app15031552.

[16] W. Y. Ayele, "Adapting CRISP-DM for Idea Mining A Data Mining Process for Generating Ideas using a Textual Dataset," 2020. [Online]. Available: www.ijacsa.thesai.org

[17] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, SCITEPRESS - Science and Technology Publications, 2018, pp. 108–116. doi: 10.5220/0006639801080116.

[18] M. Tahir, A. Abdullah, N. Izura, and K. Azhar, "DeepImputeIDS: Enhancing Intrusion Detection Systems with Deep Learning-Based Missing Data Imputation," in *2023 14th International Conference on Information & Communication Technology and System (ICTS)*, IEEE, Oct. 2023, pp. 289–295. doi: 10.1109/ICTS58770.2023.10330831.

[19] S. Wanjau and G. Kamau, "Ensemble Feature Selection for Network Intrusion Detection: Combining Information Gain and Random Forest with Recursive Feature Elimination," *International Journal of Computer and Information Technology(2279-0764)*, vol. 13, no. 4, Jan. 2025, doi: 10.24203/262j5109.

[20] V. Shanmugam, R. Razavi-Far, and E. Hallaji, "Addressing Class Imbalance in Intrusion Detection: A Comprehensive Evaluation of Machine Learning Approaches," *Electronics (Basel)*, vol. 14, no. 1, p. 69, Dec. 2024, doi: 10.3390/electronics14010069.

[21] S. Ngwenya and T. C. Garidzira, "Enhanced Fake News Detection with Domain-Specific Word Embeddings: A TorchText-Based Method for News Semantics Representation.," *The Indonesian Journal of Computer Science*, vol. 14, no. 4, Jul. 2025, doi: 10.33022/ijcs.v14i4.4831.

[22] M. Sajid *et al.*, "Enhancing intrusion detection: a hybrid machine and deep learning approach," *Journal of Cloud Computing*, vol. 13, no. 1, p. 123, Jul. 2024, doi: 10.1186/s13677-024-00685-x.

[23] V. R. Joseph and A. Vakayil, "SPlit: An Optimal Method for Data Splitting," *Technometrics*, vol. 64, no. 2, pp. 166–176, Apr. 2022, doi: 10.1080/00401706.2021.1921037.

[24] F. Maleki, N. Muthukrishnan, K. Ovens, C. Reinhold, and R. Forghani, "Machine Learning Algorithm Validation: From Essentials to Advanced Applications and Implications for Regulatory Certification and Deployment," *Neuroimaging Clin N Am*, vol. 30, no. 4, pp. 433–445, Nov. 2020, doi: 10.1016/J.NIC.2020.08.004.

[25] Z. Bami, A. Behnampour, and H. Doosti, "A New Flexible Train-Test Split Algorithm, an approach for choosing among the Hold-out, K-fold cross-validation, and Hold-out iteration," Jan. 2025, doi: arXiv:2501.06492v1.

[26] Z. Liu, Z. Xu, J. Jin, Z. Shen, and T. Darrell, "Dropout Reduces Underfitting," 2023. [Online]. Available: https://github.

[27] E. Rojas, A. N. Kahira, E. Meneses, L. B. Gomez, and R. M. Badia, "A Study of Checkpointing in Large Scale Training of Deep Neural Networks," Mar. 2021, [Online]. Available: http://arxiv.org/abs/2012.00825

[28] T.-H. Chua and I. Salam, "Evaluation of Machine Learning Algorithms in Network-Based Intrusion Detection System," Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.05232

[29] N. A. Al-Shukaili, M. L. M. Kiah, and I. Ahmedy, "Optimizing feature selection and deep learning techniques for precise detection of low-rate distributed denial of service (LDDoS) attack," *Discover Internet of Things*, vol. 5, no. 1, p. 80, Jul. 2025, doi: 10.1007/s43926-025-00182-w.

[30] M. Zakariah, S. A. AlQahtani, A. M. Alawwad, and A. A. Alotaibi, "Intrusion Detection System with Customized Machine Learning Techniques for NSL-KDD Dataset," *Computers, Materials & Continua*, vol. 77, no. 3, pp. 4025–4054, 2023, doi: 10.32604/cmc.2023.043752.

[31] D. Justus, J. Brennan, S. Bonner, and A. S. McGough, "Predicting the Computational Cost of Deep Learning Models," in *2018 IEEE International Conference on Big Data (Big Data)*, IEEE, Dec. 2018, pp. 3873–3882. doi: 10.1109/BigData.2018.8622396.

[32] F. Çolhak, H. Coşkun, T. N. R. Cyrille, T. Hoxa, M. İ. Ecevit, and M. N. Aydın, "Accelerating IoV Intrusion Detection: Benchmarking GPU-Accelerated vs CPU-Based ML Libraries," Apr. 2025, [Online]. Available: http://arxiv.org/abs/2504.01905

[33] I. Svogor, C. Eichenberger, M. Spanring, M. Neun, and M. Kopp, "Profiling and Improving the PyTorch Dataloader for high-latency Storage: A Technical Report," Dec. 2022, doi: arXiv:2211.04908v2.

[34] G. Kumar, "An improved ensemble approach for effective intrusion detection.," *Journal of supercomputing*, vol. 76, no. 1, 2020.

Sustainability and Social Impact

Offline-First Mobile Application for Communal Livestock Management: A Participatory Design Approach in South Africa's Eastern Cape

1st Oluwatobi Aduradola
Department of Computational Sciences
University of Fort Hare
Alice, South Africa
201925685@ufh.ac.za

2nd Caroline Gurajena
Department of Computational Sciences
University of Fort Hare
Alice, South Africa
cgurajena@ufh.ac.za

3rd Dare Fatumo
Department of Business and Innovation
University of Fort Hare
Alice, South Africa
ebenfatumo@gmail.com

***Abstract*- Information and Communication Technology (ICT) has emerged as a critical tool for agricultural development in developing countries, yet its implementation in communal farming areas remains challenging. This study investigates the information needs of communal livestock farmers in the Eastern Cape Province of South Africa. It presents the development of a mobile-based livestock breeding management system to address these needs. Using a mixed-methods approach, we surveyed 120 communal farmers across four locations in the Raymond Mhlaba Municipality to understand their information requirements and technology adoption patterns. The research revealed significant gaps in timely access to breeding information, with 78% of farmers relying solely on extension officers who visit infrequently. Through a participatory design process involving 15 farmers, we developed an offline Android application using React Native and Realm database that enables farmers to record sheep breeding data, track breeding cycles, and receive automated notifications for critical breeding events. The system was tested with 45 farmers over six months, showing improved record-keeping accuracy (34% to 89%) and breeding success rates (15% increase in conception rates). This paper contributes to the growing knowledge on ICT adoption in agricultural development by demonstrating how mobile technology can bridge information gaps in resource-constrained environments. It indicates that an offline-first, participatory-designed mobile application can significantly improve breeding record-keeping (from 34% to 89%) and conception rates (by 15%) for communal sheep farmers in South Africa, providing a validated model for ICT4D interventions in low-connectivity environments.**

Keywords: ICT for Development (ICT4D), Mobile Agriculture, Communal farmers, Livestock Management, Agricultural information systems, Prototyping, South Africa

I. INTRODUCTION

Livestock production is a critical component of the socio-economic fabric for millions in developing countries, supporting livelihoods and food security for over 600 million poor farmers globally. [1], [2]. In South Africa, nearly 35% of the population resides in communal areas, with a majority engaged in farming [3], [4]. Communal farmers, who constitute a significant portion of the population in developing nations, often lack access to timely and up-to-date information, hindering their ability to optimize agricultural yields [5], [6]. However, these communal farmers usually operate in isolation, with limited reach to extension officers and agricultural support services [7], [8].

A primary challenge in communal livestock management is the low reproductive rate compared to commercial operations, mainly due to an information deficit [9]. Farmers lack timely access to knowledge on best practices for breeding, nutrition, disease control, and market readiness [9]. While Information and Communication Technologies (ICTs) are recognized as powerful tools for bridging this information gap, existing solutions often require stable internet connectivity, a resource scarce in rural Eastern Cape, and are designed for high-literacy, commercial contexts, making them unsuitable for communal farmers [10], [11]. Mobile technology offers a promising solution to bridge this information gap. With the widespread availability of mobile devices, even in rural settings, there is an opportunity to deliver targeted information in an accessible and user-friendly manner [12]. However, existing agricultural applications often fail to meet the specific needs of communal farmers, frequently requiring stable internet connections or featuring complex interfaces unsuitable for users with varying levels of technological literacy [13].

This paper investigates using a mobile-based database system, an offline-first application, to improve livestock breeding management. The study had three primary objectives: to identify and understand the specific information and technological challenges faced by communal farmers in the Eastern Cape; to co-design and develop an Android application through a participatory process that provides offline record-keeping, breeding cycle tracking, and proactive information dissemination via push notifications; and to evaluate the application's usability with the target audience. This paper contributes a validated design framework and functional prototype for offline agricultural mobile applications, demonstrating their viability and impact for low-resource farming communities.

II. LITERATURE REVIEW

A. ICTs in Agricultural Development

Information and Communication Technologies (ICTs) encompass tools that facilitate information gathering, processing, storage, and dissemination [14]. ICTs have been leveraged in agriculture for various applications, including weather forecasting, market price access, extension services, and precision farm management [15], [16]. The potential of ICT to revolutionize agriculture in developing regions is well-documented, as it can facilitate knowledge sharing, improve access to markets, and enable better farm management decisions [16].

Livestock production is essential in communal farmers' socio-economic ascent, directly impacting the livelihoods of over 600 million poor farmers in developing countries [1]. The availability of timely, authentic, and relevant information is crucial for the sustainable production of

livestock in communal farming systems [3], [5]. Farmers can significantly improve their management and production skills in coping with challenges like natural disasters, diseases, and market access if they can easily access information[10]. However, the adoption of these technologies in developing countries has been inconsistent. While ICTs are a powerful tool for collecting and sharing information between farmers and stakeholders [15], many of these tools have failed to meet user expectations. Thus, farmers do not extensively utilize them [11], often due to a mismatch between design and context.

In South Africa, the government has initiated programs such as the National Agricultural Innovation System (NAIS) and e-Agriculture Strategy to integrate digital tools into farming [17]. Despite these efforts, implementation in communal areas remains limited due to persistent barriers, including poor connectivity, low digital literacy, and a lack of localized content [18].

B. Critical Gap in Existing Livestock Management Applications

Several mobile applications support livestock management globally, including iCow (Kenya), Cow Manager (India), and FarmLogs (USA). These tools include breeding tracking, health monitoring, and feed planning. Closer to this study's focus, applications like Livestocked offer robust online herd management features, HerdBoss focuses on pedigree tracking for market value, and The Lambing Book provides simple lamb record-keeping.

However, a critical analysis of these existing applications reveals a collective failure to address the core constraints of communal farmers in South Africa:

1. Online Dependency: Most, like Livestocked and HerdBoss, require a stable internet connection for core functionality, ignoring the connectivity challenges of rural areas identified in our requirement elicitation (66.7% no internet access). This reliance on connectivity is a common flaw in agricultural apps, ignoring the challenges in rural areas, a problem that necessitates the 'offline-first' architecture seen in more context-aware systems like AgriLink [19]
2. Complexity and Context Mismatch: They are often designed for high-literacy, commercial contexts, making them unsuitable for communal farmers with lower technological proficiency. Their focus on market value (HerdBoss) or complex invoicing (Livestocked) does not align with the immediate breeding management needs identified.
3. Passive Information Delivery: They primarily function as databases, lacking the proactive, time-sensitive information dissemination via push notifications crucial for managing breeding cycles effectively.
4. Lack of Offline-First Design: Few systems are architected to function primarily offline, with local data storage and synchronization capabilities.

This gap in the literature and the market justifies the introduction of this research: to develop a mobile-based system that helps a communal farmer with livestock breeding management by being offline-first, simple, proactive, and built on directly elicited user requirements.

Collectively, these gaps, online dependency, context mismatch, and passive design highlight a systemic failure in existing solutions to address communal farmers' offline, low-literacy, time-sensitive needs. Our study addresses this triad of failures by prioritizing offline-first architecture, ultra-simple UI/UX derived from participatory design, and proactive push notifications, filling a critical void in literature.

C. Mobile Technology as a Channel for Dissemination

Due to their widespread penetration across Africa, mobile phones are among the most effective channels for delivering information to rural communities [20]. Their mobility, affordability, and increasing capability make them ideal platforms for bridging the information gap. Mobile technology has emerged as an up-and-coming tool for livestock management due to its portability, decreasing costs, and increasing penetration rates in rural areas[16]. Mobile applications for livestock management typically focus on record-keeping, health monitoring, breeding management, and market information services [16].

Existing mobile applications in livestock management include platforms like Livestocked, which assists in herd management and pasture tracking, and HerdBoss, which focuses on detailed animal record-keeping. However, these applications are primarily designed for commercial farmers in developed countries and often require stable internet connections and sophisticated technical knowledge.

The unique requirements of communal farmers in developing countries necessitate different approaches to mobile application design. These include offline functionality, simplified user interfaces, local language support, and integration with existing farming practices and social structures. Mobile applications are software programs designed to run on these devices, ranging from entertainment to complex transactional systems [6], [13]. They are typically categorized as:

1. Web-based: Hosted on a website and accessed through a mobile browser.
2. Native: Built for a specific operating system (e.g., Android, iOS), offering high performance and full feature access.
3. Hybrid: Combining elements of both web and native applications.

Android OS dominates the South African market with over 70% share [21], making it the most strategic platform for targeting a large user base. These technologies' core function aligns with the information management principle in agriculture. This process allows farmers to access information and derive value from it for decision-making [22].

D. Information Needs of Communal Farmers

Understanding the specific information needs of communal farmers is crucial for developing an effective ICT solution [22]. Research has identified several key information areas critical for communal farmers, including breeding management, animal health, market prices,

weather information, and government support programs [11], [22].

Breeding management information is vital for livestock farmers as it directly impacts productivity and income generation [23]. Critical breeding information includes optimal breeding times, pregnancy monitoring, vaccination schedules, and nutrition requirements during different stages of the breeding cycle [21].

Traditional information sources for communal farmers include extension officers, other farmers, local veterinarians, and community leaders [24]. However, these sources often provide information that is not timely, standardized, or easily accessible when needed for decision-making [25].

E. Technology Adoption in Rural Communities

Adopting new technologies in rural communities follows complex social, economic, and cultural patterns. The Technology Acceptance Model (TAM) and its extensions provide frameworks for understanding technology adoption, emphasizing perceived usefulness, ease of use, and social influences [26].

In the context of agricultural technology adoption in Africa, research has identified additional factors, including compatibility with existing practices, observability of benefits, and support from change agents such as extension officers and community leaders [26]. The design of ICT solutions for rural communities must consider these factors to ensure successful adoption and sustained use.

III. METHODOLOGY

The prototyping methodology is particularly effective in ICT for Development (ICT4D) contexts, where user needs are dynamic and often poorly understood at the outset [26]. This approach allows for iterative feedback from stakeholders, ensuring the final product is refined and aligns with real-world requirements.

In this study, prototyping enabled the continuous incorporation of farmer input, which was essential for enhancing the usability and relevance of the application for its intended low-resource, low-connectivity environment.

A. Research Design

Adhering to this iterative philosophy, this paper adopts a prototyping methodology, which allows refinement based on stakeholder feedback. This paper employed a mixed-methods approach combining quantitative surveys and qualitative interviews to understand farmers' information needs, followed by a participatory design process for system development and testing. The research was conducted in four phases: requirement gathering, system design and development, pilot testing, and evaluation.

As illustrated in Figure 1, the research followed a four-phase iterative prototyping approach: Requirement Gathering, System Design & Development, Pilot Testing, and Evaluation. Feedback from each phase was used to refine the application and the research process itself.

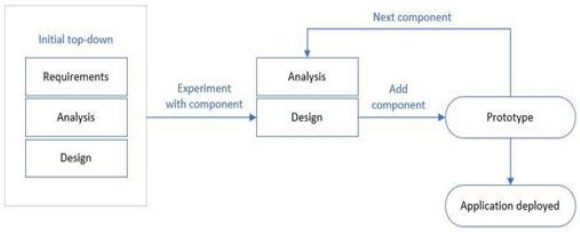


Fig. 1. The iterative prototyping methodology adopted for the development and evaluation of the mobile application

B. Study Area and Participants

The study was conducted in the Raymond Mhlaba Municipality in the Eastern Cape Province, South Africa. This area was selected due to its significant communal farming population and the researchers' established relationships with local communities and extension services.

Four locations were purposively selected: Seymour, Bedford, Adelaide, and Fort Beaufort. These locations represent diverse geographical and socio-economic conditions within the municipality. One hundred and twenty farmers participated in the initial survey, with 45 participating in the six-month pilot testing phase.

Inclusion criteria for participants included: ownership of sheep (minimum 10 animals), residence in the study area for at least two years, and willingness to participate in technology testing. The study received ethical clearance from the institutional review board, and all participants provided informed consent. Participant anonymity was ensured by assigning unique ID codes to survey and interview responses. All digital data was stored on password-protected servers, and raw interview recordings were deleted after transcription.

C. Data Collection Methods

From the initial pool of 120 surveyed farmers, a purposive sub-sample of 30 participants was selected for in-depth, semi-structured interviews. Selection criteria ensured diversity in age, gender, and farming experience. Furthermore, 15 farmers from this interview group were invited to participate in participatory design workshops, chosen for their willingness to engage and provide detailed feedback on interface concepts.

1) Quantitative Survey

A structured questionnaire was administered to 120 farmers to assess their information needs, current information sources, technology access and use, and demographic characteristics. The questionnaire was developed based on a literature review and pre-tested with 10 farmers before full implementation.

Key variables measured included:

- Demographic characteristics (age, gender, education level, farming experience)
- Flock size and management practices
- Information needs and sources
- Mobile phone ownership and usage patterns
- Attitudes towards technological adoption

2) Qualitative Interviews

Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2025

Semi-structured interviews were conducted with 30 farmers, five extension officers, and three veterinarians to gain deeper insights into information needs, communication patterns, and potential barriers to technology adoption. Interviews were conducted in local languages (Xhosa and English) and lasted 30-60 minutes.

3) Participatory Design Sessions

Participatory design sessions were conducted with 15 farmers to involve end-users in the system design process. These sessions helped identify user interface requirements, workflow preferences, and critical functionalities from the farmers' perspective.

D. System Development Approach

The application was developed using an iterative prototyping methodology to incorporate continuous user feedback.

1) Technology Stack

The application was developed using the React Native framework for cross-platform compatibility, with the Realm database for offline data storage and synchronization. This technology combination was selected based on requirements for offline functionality, cross-platform deployment, and ease of development.

Based on the requirements, a native Android application was developed. An Android-native approach was selected due to Android's market share exceeding 70% in South Africa [21]. The React Native framework was chosen for its cross-platform potential and development efficiency. For offline data storage, RealmDB was selected as our mobile database, designed explicitly for offline functionality, which meets our core requirement. The system was designed on an offline-first architecture. All data (animal records, breeding events) is stored locally on the device using RealmDB. A synchronization module was implemented to push/pull data to a central server only when internet connectivity is available, ensuring continuous operation. The user interface was designed to be simple, with large buttons and straightforward navigation to ensure usability for users with low digital literacy.

2) Development Process

The development process followed standard software engineering practices, including requirements analysis, system design, implementation, testing, and deployment. User acceptance testing was conducted iteratively throughout the development process to ensure alignment with user needs and expectations.

E. Evaluation Methods

1) Pilot Testing

A six-month pilot test was conducted with 45 farmers to evaluate system effectiveness and user acceptance. Farmers were provided with smartphones loaded with the application and received basic training on its use.

2) Data Collection During Pilot Test

Data collected during the pilot test included:

- Usage statistics (frequency of use, features accessed)
- Record-keeping accuracy (compared to traditional methods)
- Breeding performance indicators (conception rates, lambing success)
- User satisfaction and perceived usefulness
- Technical issues and challenges encountered

F. Results from Data Collection

1) Farmers' Information Needs and Current Practices

The survey revealed significant gaps in farmers' access to timely and relevant information. Key findings are shown in the table below:

TABLE I. DEMOGRAPHIC PROFILE OF SURVEY PARTICIPANTS

Characteristic	Value / Distribution
Gender	Male: 68%
Average Age	48.3 years
Educational Level	Primary:43%, Some Secondary:31%
Primary Info Source	Extension Officers (78%)
Mobile Phone Ownership	91% owned a mobile device
Smartphone Ownership	34% owned a smartphone
Internet Access	66.7% report no reliable access
Record-Keeping	34% maintain written records

Demographics: Most respondents were male (68%), aged 35 to 65 (48.3 years). Education levels were generally low, with 42% having completed primary education and 31% having some secondary education.

Information Sources: Extension officers were the primary information source for 78% of farmers, despite irregular visit patterns (average of 2-3 visits per year). Other sources included fellow farmers (56%), radio programs (34%), and veterinarians (23%).

Information Needs: Priority information needs identified were:

Breeding timing and management (89% of respondents)

- Animal health and vaccination schedules (83%)
- Market prices and selling opportunities (76%)
- Weather and seasonal planning (65%)
- Government support programs (58%)

Technology Access: Mobile phone ownership was high (91%), but smartphone ownership was limited (34%). Internet access was sporadic, with 66.7% reporting no consistent internet connectivity.

Record-Keeping Practices: Only 34% of farmers maintained written records, most relying on memory for breeding and health information. This contributed to poor breeding management and missed opportunities for productivity improvements.

Internet Access: A significant majority (66.7%) indicated they did not have reliable internet access on their mobile devices. This made offline capability a mandatory requirement.

Application Usage: 85.7% of participants had never used a mobile application to manage their livestock.

Desired Features: There was a strong preference for receiving livestock breeding information conveniently (95.2%) and for a system that could assist with record-keeping (90.5%). All participants expressed the need for

push notifications for reminders and a feature to easily contact a veterinary doctor or extension officer.

IV. System design and development

Based on the needs assessment, a mobile-based livestock breeding management system, 'AddSheep', was developed. The name reflects its core functionality of adding sheep to a digital flock. The application featured the following key functionalities:

A. Core Functionalities

- Animal Registration: Farmers can register individual sheep with details including tag number, breed, date of birth, weight, and other identifying characteristics.
- Breeding Cycle Management: The system lets farmers record breeding events, track pregnancy status, and receive automated reminders for critical breeding milestones.
- Health Monitoring: Basic health records can be maintained, including vaccination dates, treatment history, and health observations.
- Notification System: Push notifications provide timely reminders for breeding activities, vaccinations, and other important events. Automatically schedule and display alerts for key events (pregnancy signs, nutrition, vaccination, lambing) based on the breeding date.
- Contact Directory: Integration with extension officer and veterinarian contact information for easy access to expert support.
- Data Export: Records can be exported as CSV files for sharing with extension services or other stakeholders.

B. User Interface Design

- The user interface was designed with specific consideration for the target user group's characteristics:
- Large, clear fonts suitable for users with varying literacy levels.
- Icon-based navigation with text labels in local language (Xhosa).
- Simplified workflows with minimal steps required for everyday tasks
- Offline functionality with data synchronization when connectivity is available
- Voice input options for users with limited typing skills

C. Pilot Test Results

1) User Adoption and Usage Patterns

Of the 45 farmers participating in the pilot test, 38 (84%) used the application regularly throughout the six months. Usage patterns showed:

- Average of 4.2 interactions per week per user
- Most common activities: animal registration (78% of sessions), breeding record entry (65%), and notification viewing (89%)
- Highest usage periods coincide with traditional breeding seasons (March-May)

2) Impact on Breeding Management

Significant improvements were observed in breeding management practices

TABLE II. Impact of AddSheep on Breeding Management

Performance Indicator	Baseline (Pre-Pilot)	Result (Post-Pilot)
Accurate Record-Keeping	34% of farmers	89% of farmers
Breeding Conception Rate	72% (average)	87% (average)
User Adoption	~0%	84% regular usage

- Record-Keeping Accuracy: The percentage of farmers maintaining accurate breeding records increased from 34% (baseline) to 89% during the pilot period.
- Breeding Success Rates: Conception rates improved from an average of 72% (baseline) to 87% during the pilot period, representing a 15% increase in breeding efficiency.
- Timely Interventions: Farmers reported 67% improvement in timely response to breeding-related events due to automated notifications.

While these improvements strongly correlated with the adoption of AddSheep, and qualitative feedback from farmers attributed their success to the app's reminders and record-keeping, we acknowledge that other unmeasured factors (e.g., seasonal variations, concurrent training) may have also contributed to these outcomes. These include the Hawthorne effect, where participants may have improved their management practices due to the attention from the study itself, as well as seasonal variations in weather and pasture quality. A controlled longitudinal study is recommended to isolate the application's causal impact.

3) User Satisfaction and Acceptance

User satisfaction was measured through structured interviews and standardized questionnaires:

- 91% of users rated the application as "useful" or "very useful". The high perceived usefulness (91%) and willingness to continue use (85%) align with the Technology Acceptance Model [26], confirming that when a system is designed to solve a real problem (perceived usefulness) and is easy to use (addressed via participatory design), adoption follows, even in low-tech contexts.
- 78% indicated they would recommend the system to other farmers
- 85% expressed willingness to continue using the system beyond the pilot period
- Main challenges reported were the initial learning curve (43% of users) and occasional technical glitches (23%)

4) Technical Performance

The system demonstrated robust technical performance throughout the pilot period:

- Application stability: 97.8% uptime during testing period
- Data accuracy: 99.2% data integrity across all records
- Offline functionality: Successfully operated without internet connectivity for an average period of 12 days
- Battery efficiency: Application used less than 9% of daily battery capacity on average

D. Challenges and Limitations

Several challenges were encountered during the pilot implementation:

1) Technical Challenges:

- Smartphone hardware limitations among some users required system optimization.
- Initial data migration from traditional record-keeping systems was time-consuming.
- Occasional synchronization issues when internet connectivity was restored.

2) User-Related Challenges:

- Learning curve for users with limited smartphone experience
- Resistance from some older farmers to adopt new technology
- Need for ongoing technical support and troubleshooting

3) Infrastructure Limitations:

- Inconsistent electricity supply for device charging
- Poor cellular coverage in some remote areas
- Limited local technical support capacity

V. DISCUSSION

A. Implications for ICT Adoption in Agriculture

This study demonstrates the potential for mobile technology to address critical information gaps in communal farming systems. The significant improvements in record-keeping accuracy and breeding success rates suggest that well-designed ICT solutions can have measurable impacts on agricultural productivity, even in resource-constrained environments.

The high user acceptance rates (84% regular usage) indicate that communal farmers are willing to adopt new technologies when they address real needs and are designed with user capabilities in mind. This challenges assumptions about technology resistance in rural communities and suggests that the key barriers to adoption are often related to design and implementation rather than fundamental resistance to change.

Our success directly addresses the 'Online Dependency' gap [19] through offline-first design, the 'Complexity and Context Mismatch' through participatory UI development, and the 'Passive Information Delivery' gap through automated, time-sensitive push notifications.

B. Design Considerations for Rural ICT Solutions

Several design principles emerged from this research that may apply to other ICT initiatives in rural areas:

- Offline-First Design: The ability to function without internet connectivity was critical for user acceptance. Rural areas often have unreliable connectivity, making offline functionality essential for practical utility.
- User-Centered Design: Extensive user involvement in the design process was crucial for developing appropriate interfaces and workflows. Traditional software design approaches may not be suitable for users with limited technical experience.
- Contextual Relevance: The system's focus on specific, locally relevant needs (sheep breeding management) contributed to its acceptance and effectiveness. Generic solutions may be less effective than targeted applications.
- Integration with Existing Systems: The system's compatibility with existing farming practices and social structures facilitated adoption. Disruptive technologies may face greater resistance than those that complement existing practices.

C. Sustainability and Scalability

The long-term sustainability of ICT solutions in rural areas requires consideration of multiple factors:

- Economic Sustainability: The cost of smartphones and data services remains a barrier for many farmers. Subsidized programs or alternative funding mechanisms may be necessary for widespread adoption.
- Technical Sustainability: Ongoing technical support and system maintenance require local capacity development. Training programs for local technicians and extension officers are essential.
- Institutional Sustainability: Integration with existing agricultural extension systems and government programs can provide institutional support for continued system use and development.

D. Comparison with Similar Initiatives

This research contributes to the growing body of evidence on mobile technology applications in agriculture. Compared to other initiatives, this study demonstrates several unique aspects:

- Focus on offline functionality addresses a critical gap in many existing solutions
- Extensive user involvement in the design process resulted in higher acceptance rates
- Measurable impact on breeding outcomes provides concrete evidence of effectiveness

- Six-month pilot period provides insights into sustained usage patterns

E. *Limitations and Future Research*

Several limitations should be noted:

- Sample Size and Scope: The study was conducted in a single municipality with a relatively small sample size. Broader studies across different regions and farming systems would strengthen the generalizability of findings.
- Long-term Impact: While providing initial insights, the six-month pilot period may not capture long-term adoption patterns and sustained impacts.
- Economic Analysis: This study did not include a detailed economic analysis of costs and benefits, which would be valuable for policy and implementation decisions.
- Study Effect: The observed improvements may be partially influenced by the Hawthorne effect, where participants altered their behavior (e.g., paid more attention to breeding) because they were aware of being part of a study.
- Selection Bias: The participants were self-selected to some degree, as they owned phones, were willing to participate in a tech study, and met the minimum flock size criteria. This may limit the generalizability of our high adoption rates to the entire communal farmer population, which is likely to include individuals with even lower technological affinity.
- Generalizability: This study was conducted in one municipality in the Eastern Cape. While the challenges of connectivity and literacy are common across rural Africa, specific cultural practices, livestock breeds, and farming structures may differ. Testing the framework in other regions is necessary to confirm its broad applicability.

Future research should focus on:

- Long-term longitudinal studies to assess sustained impact and adoption
- Economic analysis of ICT investment returns for communal farmers
- Integration with other agricultural technologies and services
- Scaling strategies for broader implementation
- Comparative studies across different agricultural systems and regions
- Multilingual Support: Integrate Xhosa, Zulu, and Afrikaans for broader accessibility.
- Livestock Theft Prevention: Explore integration with GPS tagging or community alert systems.

VI. CONCLUSION

This research demonstrates that offline-first mobile applications, designed through participatory processes, can effectively address information gaps in communal livestock farming. The "AddSheep" system achieved high user adoption (84%) and correlated with improved breeding management outcomes. Key success factors included offline functionality, user-centered design, and contextual relevance to local farming practices.

The study contributes validated design principles for rural ICT solutions and provides evidence that communal farmers will adopt technology when it addresses real needs and is appropriately designed for their context. These findings have implications for agricultural extension programs, ICT4D initiatives, and rural development strategies in similar contexts.

While acknowledging limitations related to study scope and potential confounding factors, this research provides a foundation for scaling offline-first agricultural applications in resource-constrained environments and demonstrates the potential for ICT to contribute to improved agricultural outcomes in developing regions.

VII. ACKNOWLEDGMENTS

We thank the farmers of Raymond Mhlaba Municipality who participated in this research, the extension officers who facilitated access to farming communities, and the institutional review board for ethical guidance.

REFERENCES

[1] N. Khan, S. Fahad, M. Naushad, and S. Faisal, “Analysis of Livelihood in the World and Its Impact on World Economy,” Oct. 22, 2020, Social Science Research Network, Rochester, NY: 3717265. doi: 10.2139/ssrn.3717265.

[2] M. Herrero et al., “The roles of livestock in developing countries,” Animal, vol. 7, no. s1, pp. 3–18, 2013.

[3] V. Mmbengwa, B. Nyhodo, L. Myeki, X. Ngethu, and H. van Schalkwyk, “Communal livestock farming in South Africa: Does this farming system create jobs for poverty stricken rural areas,” Sylwan, vol. 159, no. 10, pp. 176–192, 2015.

[4] B. Cousins, A. Dubb, D. Hornby, and F. Mtero, “Social reproduction of ‘classes of labor’ in the rural areas of South Africa: Contradictions and contestations,” in Agrarian Marxism, Routledge, 2020, pp. 208–233. Accessed: Sept. 11, 2025. [Online]. Available: https://www.taylorfrancis.com/chapters/edit/10.4324/9780429197444-10/social-reproduction-classes-labour-rural-areas-south-africa-contradictions-contestations-ben-cousins-alex-dubb-donna-hornby-farai-mtero

[5] K. Mwantimwa, “Livelihood information and knowledge needs, access, and exchange in rural communities in the Bunda District, Tanzania,” Rural Soc., vol. 29, no. 1, pp. 30–43, Jan. 2020, doi: 10.1080/10371656.2020.1744271.

[6] M. Kamal and T. A. Bablu, “Mobile applications empowering smallholder farmers: A review of the

impact on agricultural development," Int. J. Soc. Anal., vol. 8, pp. 36–50, 2023.
[7] T. A. Tran, B. R. Cook, and V. Touch, "Agricultural extension institutions in rural Cambodia: Unpacking extension agent-farmer relations and interactions," J. Rural Stud., vol. 117, p. 103671, 2025.
[8] S. Maulu, O. J. Hasimuna, B. Mutale, J. Mphande, and E. Siankwilimba, "Enhancing the role of rural agricultural extension programs in poverty alleviation: A review," Cogent Food Agric., vol. 7, no. 1, p. 1886663, Jan. 2021, doi: 10.1080/23311932.2021.1886663.
[9] L. M. Molieleng, "Constraints and opportunities for increased livestock production in the communal areas of Lejweleputswa district, Free State," PhD Thesis, Central University of Technology, 2022. Accessed: Sept. 11, 2025. [Online]. Available: http://ir.cut.ac.za/handle/11462/2548
[10] N. Sandhu, S. Yadav, V. Kumar Singh, and A. Kumar, "Effective crop management and modern breeding strategies to ensure higher crop productivity under direct seeded rice cultivation system: A review," Agronomy, vol. 11, no. 7, p. 1264, 2021.
[11] S. O. Sennuga, S. A. Ujoyi, J. Bamidele, S. S. Onjewu, W. I. Lai-Solarin, and A. A. Omole, "Exploring the role of smartphone apps for livestock farmers' data management extension and informed decision-making in Nigeria," Int J Probiotics Diet., vol. 3, no. 2, pp. 46–53, 2023.
[12] S. Sindakis and G. Showkat, "The digital revolution in India: bridging the gap in rural technology adoption," J. Innov. Entrep., vol. 13, no. 1, p. 29, May 2024, doi: 10.1186/s13731-024-00380-w.
[13] I. Ouédraogo, "Mobile technology and artificial intelligence for improving health literacy among underserved communities," PhD Thesis, Université de Bordeaux; Université Nazi Boni (Bobo-Dioulasso, Burkina Faso), 2024. Accessed: Sept. 11, 2025. [Online]. Available: https://theses.hal.science/tel-04574957/
[14] R. Heeks, Information and communication technology for development (ICT4D). Routledge, 2017. Accessed: Sept. 11, 2025. [Online]. Available: https://www.taylorfrancis.com/books/mono/10.4324/9781315652603/information-communication-technology-development-ict4d-richard-heeks
[15] A. Khatri et al., "Integration of ICT in agricultural extension services: A review," J. Exp. Agric. Int., vol. 46, no. 12, pp. 394–410, 2024.
[16] O. Mapiye, G. Makombe, A. Molotsi, K. Dzama, and C. Mapiye, "Information and communication technologies (ICTs): The potential for enhancing the dissemination of agricultural information and services to smallholder farmers in sub-Saharan Africa," Inf. Dev., vol. 39, no. 3, pp. 638–658, Sept. 2023, doi: 10.1177/02666669211064847.
[17] J. Degila et al., "Digital agriculture policies and strategies for innovations in the agri-food systems—cases of five West African countries," Sustainability, vol. 15, no. 12, p. 9192, 2023.
[18] H. J. Smidt and O. Jokonya, "Towards a framework to implement a digital agriculture value chain in South Africa for small-scale farmers," J. Transp. Supply Chain Manag., vol. 16, June 2022, doi: 10.4102/jtscm.v16i0.746.
[19] E. Shang and Z. Zhang, "Cultivating Change: Developing a Regenerative Agriculture Playbook for TechnoServe," PhD Thesis, Duke University, 2025. Accessed: Sept. 13, 2025. [Online]. Available: https://dukespace.lib.duke.edu/bitstreams/f9045d65-40da-46a6-8a88-971e976f508f/download
[20] J. L. van Zanden, "Examining the relationship of information and communication technology and financial access in Africa," J. Bus. Econ. Options, vol. 6, no. 3, pp. 26–36, 2023.
[21] S. Avle, "Hardware and data in the platform era: Chinese smartphones in Africa," Media Cult. Soc., vol. 44, no. 8, pp. 1473–1489, Nov. 2022, doi: 10.1177/01634437221128935.
[22] H. Mabuga, A. Sarwatt, and E. Longo, "Can Mobile Phones Bridge the Information Gap? An Assessment of Mobile Phone Usage and Agricultural Market Information Access for Smallholder Farmers in Kasulu District, Tanzania," Rural Plan. J., no. 1, pp. 95–106, 2025.
[23] S. F. U. Vanvanhossou, L. H. Dossa, and S. König, "Sustainable management of animal genetic resources to improve low-input livestock production: Insights into local Beninese cattle populations," Sustainability, vol. 13, no. 17, p. 9874, 2021.
[24] N. Zimu-Biyela, "Information needs for livestock keepers of the Dlangubo village in KwaZulu-Natal, South Africa: the role of extension officers and libraries," 2016, Accessed: Sept. 11, 2025. [Online]. Available: https://library.ifla.org/id/eprint/1476/
[25] M. Alojail, M. Alturki, and S. Bhatia Khan, "An informed decision support framework from a strategic perspective in the health sector," Information, vol. 14, no. 7, p. 363, 2023.
[26] S. A. Alabdali and S. F. Pileggi, "A Research Framework to identify determinants for Smart Technology adoption in Rural Regions," in Conference on Creativity, Technology, and Sustainability, Springer Nature Singapore Singapore, 2024, pp. 3–15. Accessed: Sept. 11, 2025. [Online]. Available: https://library.oapen.org/bitstream/handle/20.500.12657/100786/9789819785889.pdf?sequence=1#page=28

Fibre-Optic Interferometric Sensor System for Real-Time Structural Health Monitoring

Sandisiwe Bangani
Department of Physics
Nelson Mandela University
Port Elizabeth, South Africa
s215121457@mandela.ac.za

James Jena
Department of Physics
Nelson Mandela University
Port Elizabeth, South Africa
james.jena@mandela.ac.za

Victor Chike Agbakoba
Department of Physics
Nelson Mandela University
Port Elizabeth, South Africa
s215161440@mandela.ac.za

David Waswa
Department of Physics
Nelson Mandela University
Port Elizabeth, South Africa
david.waswa@mandela.ac.za

Stephen Ekolu
Civil Engineering Department
Nelson Mandela University
Port Elizabeth, South Africa
stephen.ekolu@mandela.ac.za

***Abstract*—The development of mechanical strength in concrete during curing is a crucial factor in evaluating the structural integrity and quality control of concrete. This study proposes the application of a Mach-Zehnder interferometric (MZI) based sensor system for non-destructive monitoring of the compressive strength growth of an unreinforced concrete beam. Two beams were cast, one with 60/40 sand-to-grit and the other with 40/60 sand-to-grit concentration. An optical fibre-based MZI was used to detect vibrations while the beams were curing in a water bath. The interference signal produced by the recombination of light from the sensing and reference arms was studied in the frequency domain to determine the beam's resonance frequency. A direct relationship was found between this resonance frequency and the compressive strength of the concrete, enabling a real-time estimate of strength gain during the curing process. The proposed non-destructive method estimated compressive strengths of 30.6 and 38.5 MPa for the 60/40 mixture, and 32.8 and 41.9 MPa for the 40/60 mixture at 14 and 28 days, respectively. These results compare favorably with the destructive test results of 34.1 and 37.0 MPa, and 36.1 and 38.2 MPa for the same mixtures. Beyond laboratory validation, this approach has great promise for real-world construction monitoring, where early detection of insufficient strength development can inform timely decision-making, improve quality assurance, and improve structural safety in important infrastructures.**

***Index Terms*—Mach-Zehnder Interferometer, Compressive strength, Resonance Frequency, Structural Health Monitoring**

I. Introduction

In recent years, the construction industry has seen a substantial shift toward construction automation, driven by the desire for enhanced productivity, cost effectiveness, and quality. One area that has received special attention is the increased need for precast construction, where building components like walls, floors, and beams are cast in reusable molds at a controlled off-site location, then transported to the construction site for assembly. Ensuring the quality of these precast pieces throughout the manufacturing stage is critical, demanding the tracking of strength growth in early-age concrete [1]–[3].

Concrete is the most widely used construction material due to its versatility, strength, and durability. Its structural performance, however, is heavily reliant on the development of compressive strength during the curing process [4]. Monitoring this strength progression is therefore critical for assuring safety, improving building timelines, and preventing premature loading of structures.

Traditionally, concrete strength is assessed using destructive testing methods such as cube or cylinder compression tests, which yield discrete data points and require the use of sacrificial samples. While these approaches are reliable, they are labour-intensive, time-consuming, and do not give continuous, real-time information about strength development in the structure [5]. As a result, non-destructive test methods are required to adequately assess concrete quality throughout the manufacturing and curing processes. Several non-destructive procedures have been developed to assess the compressive strength of concrete by evaluating other parameters, presenting viable alternatives to destructive testing. [6], [7].

Optical fibre sensors (OFS) have gained widespread attention due to their high sensitivity, resistance to electromagnetic interference, small size, and ability to be installed directly into concrete structures without significantly compromising structural integrity [8], [9].

OFS techniques are widely employed in concrete monitoring due to their capacity to measure strain, temperature, and identify cracks with great sensitivity and durability. Fibre Bragg Grating (FBG) sensors provide microstrain-level point monitoring of strain and temperature [10], making them ideal for embedding across various structures. Similar to FBGs, Fabry-Perot Interferometers (FPI) provide precise, local strain or temperature [11] monitoring that is appropriate

for lab-based crack and micro-deformation detection, but they are less suitable for field deployment due to the difficulty of manufacturing and multiplexing of the sensing structure. Distributed optical fibre sensing (DOFS) can detect the evolution of internal strains and changes in temperature induced by hydration reactions, shrinkage, thermal effects, and early-age microcracking, providing real-time information about material integrity and curing quality. In [12], the authors demonstrated the application of the Brillouin optical time domain reflectometry (BOTDR) in monitoring the temperature of a concrete pile during curing and the strain during loading.

A Mach-Zehnder interferometric (MZI) system excels in monitoring dynamic vibration and strain [13], [14]; hence, it has the ability to monitor a concrete structure's resonance frequency, which may then be related to the structure's compressive strength. Interferometric techniques, in particular, enable accurate detection of minute strain and vibration changes, making them effective tools for tracking the mechanical strength evolution of curing concrete and moisture content [15], [16].

The use of MZI fibre optic-based sensor systems for monitoring the development of compressive strength in unreinforced concrete beams represents an advanced non-destructive testing approach in structural health monitoring (SHM).

Despite breakthroughs in optical sensing, few studies have used interferometric techniques to monitor the early-age strength development of concrete. Most present systems for non-destructive testing of compressive strength involve ultrasonic or acoustic methods, remote monitoring systems [1], which, while effective, are frequently less sensitive to subtle changes in material stiffness [7], [17]. The use of MZI-based monitoring thus provides a potential path for real-time, non-invasive evaluation of concrete curing dynamics. By embedding MZI sensors, engineers can detect early-age shrinkage, thermal effects during curing, and load-induced stresses, enabling predictive maintenance and improved safety in civil infrastructure such as bridges and buildings [18], [19].

The purpose of this study is to investigate the usage of the MZI system to track the development of compressive strength in concrete beams as they cure. The work aims to build a robust, continuous monitoring system by linking the derived resonance frequency to compressive strength. Beyond laboratory validation, the findings of this research have the potential to enhance construction quality assurance, reduce reliance on destructive tests, and provide engineers with timely insights into safer and more efficient construction procedures.

II. Theoretical Background

The Mach-Zehnder interferometer (MZI) splits a coherent light source into two arms: sensing and reference. At the output, the optical beams recombine to generate an interference signal that is sensitive to external perturbations. The output intensity is expressed as [20]:

$$I(t) = I_s + I_r + 2\sqrt{I_s I_r}cos(\Delta\phi(t)), \tag{1}$$

where I_s and I_r are the intensities in the sensing and reference arms, respectively, and $\Delta\phi$(t) is the phase shift between them, given by

$$\Delta\phi(t) = \frac{2\pi n_{eff}}{\lambda}\Delta L(t), \tag{2}$$

with λ being the wavelength of the source signal and $\Delta L(t)$ the change in optical path length induced by strain. When the sensing fibre experiences vibrations, mechanical strain modulates $\Delta L(t)$, thereby producing measurable intensity fluctuations. The change in the optical path length due to strain is given by [21],

$$\Delta L(t) = l \times \epsilon(t) \times (1 + n_{eff}P_e), \tag{3}$$

where n is the effective refractive index of the fibre core, l is the embedded fibre length, $\epsilon(t)$ is the axial strain induced on the sensing fibre by the continuous shrinkage of the curing beam over time, and P_e is the photoelastic coefficient.
For unreinforced concrete beams, MZI sensors detect compressive strength development by monitoring strain from hydration-induced shrinkage or applied loads, as strength correlates with elastic modulus and deformation resistance [22], [23].

III. Signal Processing

The measured time-domain interference signal, I(t), contains embedded oscillations corresponding to the beam's vibration modes. However, I(t) contains a DC offset (I_s + I_r) term, which can be removed to isolate the oscillatory component as follows;

$$I_0(t) = I(t) - < I(t) >,$$

where $< I(t) >$ is the mean of the signal. Using least-squares detrending, a linear trend was removed, and we then normalise the amplitude so the signal lies within [-1, 1]:

$$I_n(t) = \frac{I_0(t)}{max|I_0(t)|} \tag{4}$$

In order to reduce high-frequency noise while keeping key features such as interference fringes, wavelet-based denoising was applied using the Symlet-4 ("sym4") wavelet up to level 4. The Bayesian soft-thresholding rule was applied to reduce the wavelet coefficients w_i [24]:

$$\hat{w}_i = \text{sign}(w_i)\ \max(|w_i| - T_B, 0)$$

where T_B is the adaptive threshold estimated using the BayesShrink method. The denoised signal is reconstructed as:

$$I_d(t) = \sum_i \hat{w}_i \psi_i(t)$$

where $\psi_i(t)$ are the wavelet basis functions (Symlet-4, up to level 4 decomposition).
A third-order Butterworth bandpass filter with normalised cutoff frequencies $[f_L, f_H]$ was created to isolate the frequency band of interest while removing undesired low and high frequency components in the denoised signal, $I_d(t)$. The filtered output $I_f(t)$ is given by the difference equation [25]:

$$I_f(t) = \sum_{k=0}^{n} b_k I_n(t-k) \; - \sum_{k=1}^{n} a_k I_f(t-k)$$

where a_k and b_k are the filter coefficients and n = 3 (filter order). To prevent phase distortion in the interferometric signal, zero-phase filtering (*filtfilt* in MATLAB) was achieved utilizing forward and reverse filtering. The filtered signal, $I_f(t)$, is thus a clean and phase-preserving version of the interferometric response, suitable for frequency-domain analysis.

Finally, the fundamental frequency (resonance frequency) f_r was determined from the spectral content of the normalised interference signal $I_n(t)$ through the Fast Fourier Transform (FFT):

$$argmax_f |\mathcal{F}\{I_f(t)\}(f)| \; \Rightarrow \; f_r$$

f_r is the frequency associated with the largest spectral peak of the interference signal.

According to the standard test method, f_r represents the fundamental transverse resonance frequency of the concrete beam [26]. Using f_r, we can obtain the dynamic Young's modulus (in GPa) of elasticity of the beam using the following equation.

$$E_d = CMf_r^2 \tag{5}$$

where C = 0.9464(L^3T/bt^3), M is the mass of the beam (in kg), b and t are the dimensions of the beam (in metres), t being the direction in which the beam is excited. T is the correction factor, which is determined by the ratio of the radius of gyration (K = t/3.464), to the length of the specimen, L.
And the following expression gives the static Young's modulus (in GPa) of elasticity of the beam for a normal weight concrete [27];

$$E_s = 4.70\sqrt{f_c} \tag{6}$$

where f_c is the compressive strength (in MPa) of the beam.
An empirical relation between elastic and dynamic Young's modulus is given as [28]:

$$E_s = 0.83E_d \tag{7}$$

Therefore, the compressive strength and the dynamic modulus can be related according to the following expression.

$$f_c = (\frac{E_d}{5.7})^2 = 0.0312E_d^2, \tag{8}$$

The compressive strength development f_c over time follows the following expression [29];

$$f_c(t) = [\frac{t}{a+bt}]f_{c,28} \tag{9}$$

where t is the age of the concrete, $f_{c,28}$ is the average compressive strength on day 28 (in MPa), a (in days), and b are constants which depend on both the cement used and the type of curing process used.

IV. Methodology

A. Specimens Preparation

Quality aggregates are essential for long-lasting concrete buildings. When evaluating aggregate quality, variables such as grading, shape, and durability should be taken into consideration. High-quality aggregates reduce voids and increase packing density, improving structural integrity. Table I gives the mix proportion of two concrete beams that were tested. And the slump test from the two mixtures. The chemical compositions and type of cement used are given in Table II, in comparison to the SANS 50197-1 requirements.

Two concrete beams (10 x 10 x 50 cm) were cast with different concentrations of sand and grit/crusher sand. Beam 1 had a 60/40 concentration of sand-grit ratio, while Beam 2 had a 40/60 sand-grit ratio. The two beams were used to investigate the development of compressive strength. While casting, two optical fibre sensors were carefully embedded in the concrete mixture at a height of 5 cm, and then concrete was poured into the mold until it was filled. The beams were placed on a vibrator to eliminate air bubbles and voids, resulting in a denser, stronger, and more uniform final product, as shown in Fig. 1.

TABLE I: Mix proportions for the two mixtures - Target strength: Moderate Strength.

Material Used	**Concrete Beam Mixtures**	
	Mixture 1 (kg)	***Mixture 2 (kg)***
River Sand	4.50	3.00
Crusher sand	3.00	4.50
Stone (19 mm)	11.4	11.4
CEM II PPC (42.5N)	4.00	4.00
Water	2.00	1.79
Slump (mm)	25	20

TABLE II: Chemical composition of the Portland Pozzolana Cement (PCC): Surebuild Premium Cement, CEM II/B-L 42.5 N

Chemical (% by mass)	SANS 50197-1 Requires	Results
Sulphur Trioxide	$\leq$ 3.5%	2.1
Chloride Content	$\leq$ 0.10%	$\leq$ 0.01

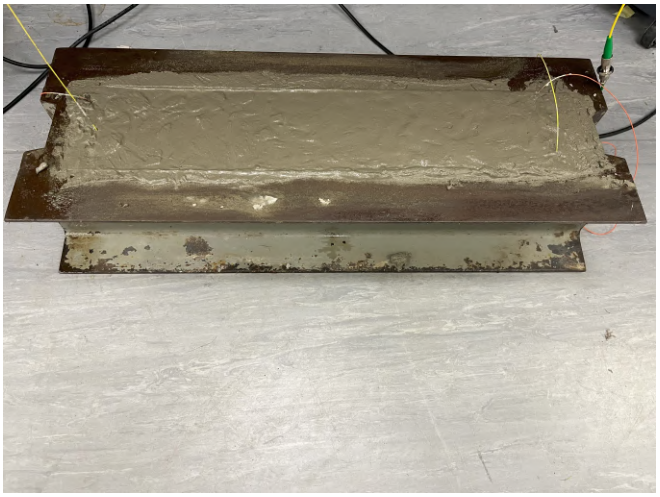


Fig. 1: Freshly cast concrete beam specimen of dimensions 100 × 100 × 500 mm with optical fibre sensors embedded.

B. Curing Conditions

After 24 h of initial curing, all specimens were demolded and submerged in a water bath to ensure proper curing until designated testing ages of 14 and 28 days for compressive strength. The curing water was maintained at temperatures between 20 - 25 °C, providing a consistent and controlled environment for hydration for a period of 28 days. Immersion curing was chosen because it prevents moisture loss, which is essential for complete hydration of cementitious materials and the development of concrete strength. This method also minimizes the risk of surface drying, shrinkage, cracking, and uneven strength development. Curing maintains proper moisture and temperature, allowing hydration - the chemical reaction between cement and water - to densify the microstructure of the concrete and minimise permeability [30]. Proper curing ensures that the concrete achieves its intended load-bearing capacity and long-term serviceability.

C. Destructive Method of Compression Strength Testing

The compressive strength of the concrete was determined in accordance with the standard destructive testing procedures. Four cube specimens of dimensions 100 ×100 ×100 mm were prepared for each mixture (60/40 and 40/60 sand-to-grit ratios). After casting, the cubes were demolded after 24 hours and cured in water until the testing ages of 14 and 28 days. At the end of the curing period, each cube was surface-dried and placed centrally between the platens of a compression testing machine, as shown in Fig. 2. A load of 180 kN/min was applied gradually until specimen failure occurred. The maximum load at failure was recorded, and the compressive strength was calculated using:

$$f_c = \frac{P}{A} \tag{10}$$



Fig. 2: Concrete cube sample (100 ×100 ×100 mm) undergoing compression strength test on the compression testing machine.

where P is the maximum load (N), and A is the cross-sectional area of the cube (mm^2). The resulting values were averaged over two specimens for each mixture and curing age.

D. Flexural Strength Testing

The flexural strength (modulus of rupture) was measured using prism specimens of dimensions 100 × 100 × 500 mm. After 24 hours of casting, the prisms were demolded and cured in a water bath until the testing age of 28 days. Testing was conducted using a three-point loading configuration, with the beam placed on two supports 300 mm apart and the load applied at the midspan, as shown in Fig. 3. A load was applied at a rate of 30 kN/min until failure, and the maximum load at failure (P) was recorded. The flexural strength was calculated using the equation [32];

$$f_b = \frac{3Pa}{bd^2} \tag{11}$$

where a - is the average distance between the line of fracture and the nearest support, b and d are the width and depth of the specimen, respectively. The resulting flexural strengths were 4.57 MPa and 4.35 MPa for the 60/40 sand-to-grit mixture and for the 40/60 sand-to-grit mixture, respectively.



Fig. 3: Flexural strength testing of the 100 × 100 × 500 mm unreinforced concrete beam.

The modulus of rupture represents the beam's resistance to bending stresses and complements the compressive strength results in assessing the overall mechanical performance of the mixtures.

V. Proposed Non-destructive Method of Compression Strength Testing

The proposed optical setup for the non-destructive testing (NDT) compression strength of the concrete beam is shown in Fig. 4. The MZ interferometric system utilizes a narrow-line-width laser (Koheras Acoustik) as the light source, which emits light at a wavelength (λ) of 1550.24 nm with an intensity of 14.79 dBm. The primary light signal was split into two arms using a 50/50 splitter. One arm was directed through a standard single-mode fibre (SMF-28, core diameter 9 μm, cladding 125 μm) that is embedded directly into the concrete beam, serving as the sensing element. The other arm forms the reference channel, isolated from environmental perturbations.
When the concrete specimen was subjected to mechanical excitation, such as a slight impact with a hammer, the embedded fibre experienced strain corresponding to the internal vibrations of the beam. These vibrations, which are governed by the internal stiffness and density of the beam, produced changes in the optical path length of the sensing arm relative to the reference arm.

The two light signals are recombined using a 2/2 coupler, and the resultant interference signal is captured by a balanced photodiode (BPD) (Optilab 22 GHz Balanced Photoreceiver Module). The output signal from the BPD was captured by the oscilloscope (Tektronix, 5 series Mixed Signal Oscilloscope), and it contains data on phase variations which directly correlate with changes in the concrete's resonance frequency.

The concrete beam's resonance frequencies were determined by studying the interference fringes in both the time and frequency domains. Given that resonance frequency is directly influenced by the material's elastic properties, the dynamic Young's modulus was calculated using the measured frequencies. The compressive strength was then estimated indirectly by comparing the dynamic modulus to typical stiffness-to-strength equations, using Equation (8).

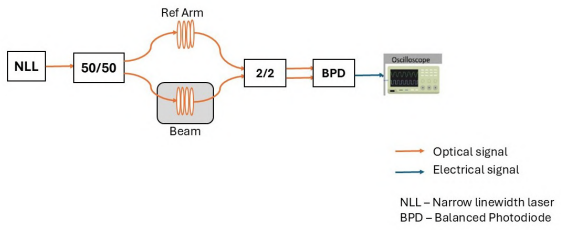


Fig. 4: Experimental setup for the non-destructive testing of the compression strength of an unreinforced concrete beam.

VI. Results and Discussion

The resonance frequency of a concrete beam is directly proportional to its stiffness and thus the dynamic modulus of elasticity. As the concrete cures, the continuing hydration of the cement causes the creation of more calcium-silicate-hydrate (C-S-H) gel, which gradually fills the pore space, densifies the microstructure, and strengthens the link between aggregates and paste [31]. This microstructural refinement causes a gradual increase in stiffness, and consequently a rise in resonance frequency over time.

The time-domain trace, Fig. 5 (a), illustrates the interference signal recorded by the MZI sensor as the concrete beam was tapped with a hammer. The impact caused the beam to vibrate, which was detected by the embedded optical fiber. The resulting trace has a transient oscillating response that gradually decays over time due to damping within the material. This decrease is typical of free vibration in concrete beams, where internal friction and microstructural heterogeneity reduce vibrational energy.

Transforming the signal into the frequency domain via the Fast Fourier transform (FFT) revealed unique resonance peaks that correspond to the beam's natural frequencies, as shown in Fig. 5 (b). The first fundamental frequency was the most dominant, as expected, because it transports the majority of the vibrational energy and is most affected by the beam's stiffness and material qualities.

A distinct fundamental frequency peak indicates the MZI system's ability to capture structural vibrations. The agreement between the time-domain decay pattern and the sharp frequency-domain peak demonstrates the optical method's robustness in non-destructive dynamic analysis of concrete beams.

Fig. 6 shows the resonant frequency over time, that is, the age of the concrete mixture. The resonant frequency is particularly low at early ages (e.g., before 7-14 days) because cement hydration is incomplete, and the concrete matrix retains a considerable fraction of unhydrated cement particles, as well as higher porosity. As a result, the beam has less stiffness and a lesser ability to transmit vibrational energy efficiently. On day 14, there is a notable increase in resonance frequency, and a gradual increase until day 28. This represents the rapid hydration reactions that occur during this time, which considerably increase the modulus of elasticity. The graph exhibits a sharp slope during this intermediate curing phase, indicating a significant gain in strength.

While hydration might last for months or even years, the gradual increase in resonance frequency (and consequently compressive strength) reduces over time. The plateau behavior is consistent with the well-known logarithmic pattern of concrete strength development. Due to resonance

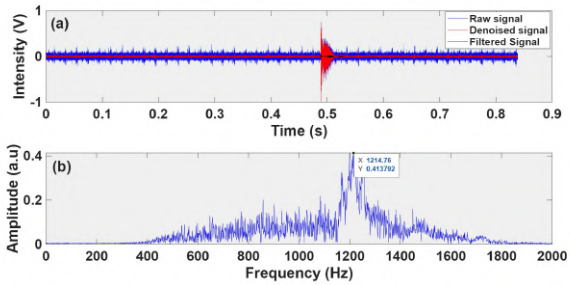


Fig. 5: (a) A time-domain signal, denoised and filtered. (b) Frequency domain signal showing a peak frequency indicating the resonance frequency.

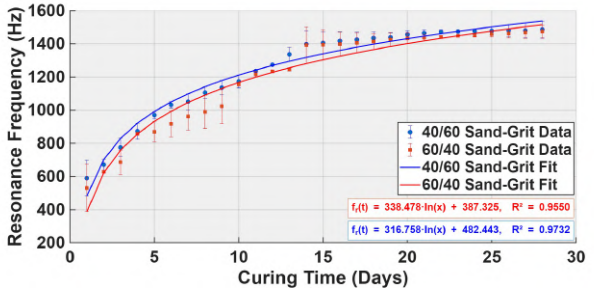


Fig. 6: Logarithmic curve displaying the evolution of the resonance frequency of the two beams with time.

frequency being proportional to the square root of the modulus of elasticity according to Equation (5), even a small increase in stiffness induce a noticeable frequency shift. This makes vibration-based non-destructive testing, such as using the MZI, extremely sensitive for monitoring cure progress and predicting compressive strength evolution.

Concrete's compressive strength develops over time through the hydration process, in which cement reacts with water to form a hardened matrix, often achieving design strength within 28 days.

Compressive strength results obtained from both destructive and non-destructive (Fig. 7) tests provide crucial information on the performance of the two concrete mixtures (60/40 sand-to-grit and 40/60 sand-to-grit) as well as the dependability of testing methods.

Table III gives a comparison of the results from the two methods, destructive and non-destructive. It can be noted that a higher proportion of coarse aggregate (grit) increases compressive strength. The likely reason is the improved interlock between particles and better load transfer pathways due to the denser packing effect of the coarse aggregates, reducing microcracking and porosity within the concrete matrix.

Between 14 and 28 days, compressive strength increased in both mixtures, indicating continuing cement hydration and microstructural densification. The non-destructive test results

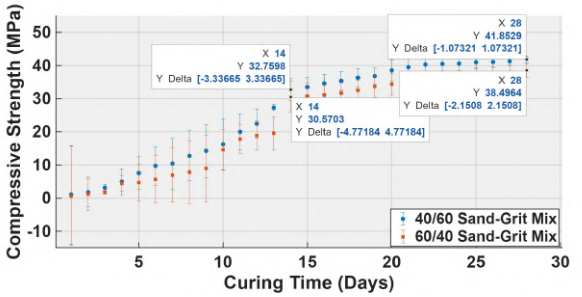


Fig. 7: Development of the compressive strength of the two beams. With days 14 and 28, compressive strengths highlighted.

TABLE III: Measured compressive strength from the two testing methods.

Age (Days)	*Mixture 1 (MPa)*		*Mixture 2 (MPa)*	
	Destructive	Non-Destr.	Destructive	Non-Destr.
14	34.1	30.6	36.1	32.8
28	37	38.5	38.2	41.9

showed a greater increase (30.6 $\rightarrow$ 38.5 MPa for the 60/40 mix and 32.8 $\rightarrow$ 41.9 MPa for the 40/60 mix), indicating that non-destructive techniques are sensitive to microstructural refinement that affects stiffness and apparent strength.

The consistent trend of increased strength in the 40/60 sand-to-grit mixture demonstrates the structural benefit of increasing coarse aggregate content in moderate-strength mixes. Furthermore, the significant agreement between destructive and non-destructive results at 28 days suggests that non-destructive testing can be a dependable option for in-situ monitoring of concrete maturation and quality.

A relationship between compressive strength (in MPa) and resonance frequency (in Hz) of the concrete beams was established using non-linear regression of experimental data, resulting in a power-law curve, as depicted in Fig. 8. Both mixtures have similar exponents of 4, but a slightly different scaling factor for the 40/60 mixture [$(8.55 \times 10^{-12} \pm 1.99 \times 10^{-19})$ MPa/Hz^4] compared to the 60/40 mixture [$(8.15 \times 10^{-12} \pm 2.35 \times 10^{-16})$ MPa/Hz^4], showing a consistent functional form with mixture-specific sensitivity. Both are fitted to power-law models with correlation coefficients R^2 = 1.00.

The exponent suggests that compressive strength varies fourth-order with resonance frequency. This is in line with the physical reality that resonance frequency is proportional to the specimen's stiffness-to-mass ratio; as concrete densifies, stiffens, and becomes more cohesive (owing to hydration and reduced porosity), both resonance frequency and compressive strength increase.

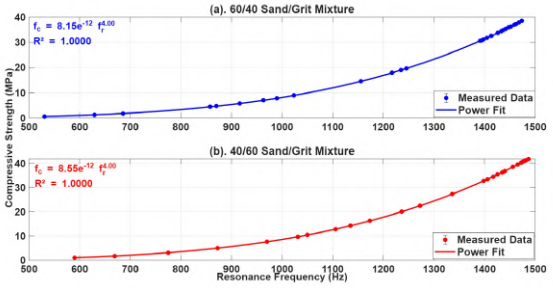


Fig. 8: Calibration curves of compressive strength against resonance frequency for the two mixtures.

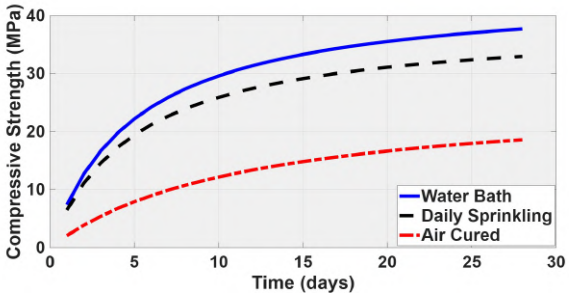


Fig. 9: Simulated compressive strength development for 3 different curing conditions: Water bath, Sprinkling, and Air-cured.

The compressive strength growth curves for concrete under the three curing methods - water bath (immersion), daily sprinkling, and air curing - differ mainly because moisture plays a role in the hydration of cement, as previously mentioned in the subsection IV-B. Fig. 9 shows a simulation of how the compressive strength develops for concrete beams cured under different conditions, in accordance with Equation (9), where a and b are 4.5 and 0.90, respectively, for water and sprinkle curing conditions. And for air curing a and b, they are 11 and 0.95, respectively.

Water bath curing (complete water immersion) provides ideal conditions by ensuring continuous moisture saturation and a constant temperature, limiting evaporation from the concrete surface. This allows for continuous hydration from the start, resulting in a denser microstructure, decreased porosity, and optimal synthesis of strength-giving chemicals. As a result, strength grows swiftly initially and reaches its maximum values at 28 days.

Daily sprinkling provides irregular moisture by periodic watering, which is superior to air curing but inferior to full immersion. It inhibits evaporation to some extent, but allows for drying between applications, resulting in partial moisture loss and irregular hydration.

Air curing exposes the concrete to ambient conditions without adding moisture, resulting in fast evaporation from the surface, particularly during the first few days when the concrete is most vulnerable. This produces dehydration, which stops or slows hydration when internal water is depleted, leading to incomplete cement reactions, increased porosity, surface cracking, and decreased long-term strength.

VII. Conclusion

The compressive strength development of two concrete mixtures with varying sand-to-grit ratios (60/40 and 40/60) was examined utilizing both destructive and non-destructive testing methods. The results showed that the 40/60 sand-to-grit mixture consistently achieved higher compressive strengths than the 60/40 mixture at 14 and 28 days, highlighting the importance of coarse aggregate content in improving load transfer, reducing porosity, and improving concrete mechanical performance.

Both testing methods validated the expected increase in strength with curing age, which is due to continuous cement hydration and microstructural densification. However, there were differences between destructive and non-destructive results: non-destructive tests tended to underestimate strength at 14 days but showed close agreement, and a slight overestimation, at 28 days. These discrepancies highlight the importance of calibrating non-destructive testing methodologies, especially during the early curing stages, when hydration is incomplete and material properties are diverse.

The combination of destructive and non-destructive testing enables a thorough evaluation of concrete performance, including precise strength determination and non-invasive monitoring for field applications.

These findings have practical applications in mixture design and quality control in construction, where nondestructive testing can reduce reliance on expensive and time-consuming destructive technologies while ensuring the safety and reliability of concrete structures.

VIII. Acknowledgment

The author would like to thank Telkom's Centre of Excellence, the CSIR, National Research Foundation (NRF), and Lambda. Special thank you to Mrs. Rozani of Nelson Mandela University, North Campus, Civil Engineering Department.

References

[1] Ranz, J., S. Aparicio, J. V. Fuente, J. J. Anaya, and M. G. Hernández. "Monitoring of the curing process in precast concrete slabs: An exper-

imental study." *Construction and Building Materials* 122 (2016): 406-416.

[2] Newell, Shane, Jamie Goggins, and Magdalena Hajdukiewicz. "Real-time monitoring to investigate structural performance of hybrid precast concrete educational buildings." *Journal of Structural Integrity and Maintenance* 1, no. 4 (2016): 147-155.

[3] Reichenbach, Sara, and Benjamin Kromoser. "State of practice of automation in precast concrete production." *Journal of Building Engineering* 43 (2021): 102527.

[4] Tulip, Shibli, Niloy Sarker, Nurun Nahar, Mijia Yang, and Ewumbua Monono. "Effect of process parameters on the physical and mechanical properties of cornstarch-based construction materials." *Construction Materials* 3, no. 3 (2023): 320-336. machine learning." *Discover Sustainability* 6, no. 1 (2025): 358.

[5] Chandam, Geetanjali, Eunjong Ahn, and Myoungsu Shin. "Effectiveness of wave interferometry in monitoring strength and stiffness development of early-age concrete." *Construction and Building Materials* 419 (2024): 135536.

[6] ACI, ACI. "228.1 R-19 Report on Methods for Estimating In-Place Concrete Strength." In *American Concrete Institute*. (2019).

[7] Saremi, Setare Ghahri, and Dimitrios Goulias. "Concrete strength gain monitoring with non-destructive methods for potential adoption in quality assurance." *Construction and Building Materials* 260 (2020): 120464.

[8] Liu, Xin, Baoquan Jin, Qing Bai, Yu Wang, Dong Wang, and Yuncai Wang. "Distributed fiber-optic sensors for vibration detection." *Sensors* 16, no. 8 (2016): 1164.

[9] Barrias, António, Joan R. Casas, and Sergi Villalba. "Embedded distributed optical fiber sensors in reinforced concrete structures—A case study." *Sensors* 18, no. 4 (2018): 980.

[10] Jo, Hye Jin, Dong Young Park, Kyung Won Kim, and Tae-Jung Ahn. "Real-time monitoring of concrete curing using fiber Bragg grating sensors: Strain and temperature measurement." *Sensors and Actuators A: Physical* 362 (2023): 114650.

[11] Zou, Xiaotian, Alice Chao, Ye Tian, Nan Wu, Hongtao Zhang, Tzu-Yang Yu, and Xingwei Wang. "An experimental study on the concrete hydration process using Fabry–Perot fiber optic temperature sensors." *Measurement* 45, no. 5 (2012): 1077-1082.

[12] De Battista, N., C. Kechavarzi, and K. Soga. "Distributed fiber optic sensors for monitoring reinforced concrete piles using Brillouin scattering." In *Sixth European Workshop on Optical Fibre Sensors*, vol. 9916, pp. 114-117. SPIE, 2016.

[13] Kurzych, Anna T., and Leszek R. Jaroszewicz. "Vibration sensing with the optical fibre Mach-Zehnder interferometer." Opto-Electronics Review (2023): e148992-e148992.

[14] Her, Shiuh-Chuan, and Chih-Min Yang. "Dynamic strain measured by Mach-Zehnder interferometric optical fiber sensors." Sensors 12, no. 3 (2012): 3314-3326.

[15] Png, W. H., H. S. Lin, C. H. Pua, J. H. Lim, S. K. Lim, Y. L. Lee, and F. A. Rahman. "Feasibility use of in-line Mach–Zehnder interferometer optical fibre sensor in lightweight foamed concrete structural beam on curvature sensing and crack monitoring." *Structural Health Monitoring* 17, no. 5 (2018): 1277-1288.

[16] Swart, Pieter L., Riaan Naude, and Beatrys M. Lacquet. "Determining the moisture content in concrete with a fibre optic Mach-Zehnder interferometer: a feasibility study." *Measurement Science and Technology* 12, no. 7 (2001): 927.

[17] Kot, Patryk, Magomed Muradov, Michaela Gkantou, George S. Kamaris, Khalid Hashim, and David Yeboah. "Recent advancements in non-destructive testing techniques for structural health monitoring." *Applied Sciences* 11, no. 6 (2021): 2750.

[18] Govindaraju, Silambarasan, and Gulshan Taj Mohammed Nabi Anwar Basha. "Real-time concrete strength monitoring: an IoT-enabled framework integrating electrochemical and fiber optic sensors for structural integrity assessment." Matéria (Rio de Janeiro) 30 (2025): e20240784.

[19] Nurzhigit Smailov, Akmaral Tolemanova, Assem Ayapbergenova, Yerlan Tashtay, and Aziskhan Amir, "Modeling and Application of Fibre Optic Sensors for Concrete Structures: A Literature Review," Civil Engineering and Architecture, Vol. 13, No. 3, pp. 1885 - 1897, (2025). DOI: 10.13189/cea.2025.130332.

[20] Miliou, Amalia. "In-fiber interferometric-based sensors: Overview and recent advances." In *Photonics*, vol. 8, no. 7, p. 265. MDPI, (2021).

[21] Zhang, Gang, Linguang Xu, and Qiang Ge. "An in-fiber Mach–Zehnder strain sensor for studying multimode interference of light." American Journal of Physics 92, no. 5 (2024): 385-391.

[22] Willberry, James Owen, Mayorkinos Papaelias, and Gerard Franklyn Fernando. "Structural health monitoring using fibre optic acoustic emission sensors." Sensors 20, no. 21 (2020): 6369.

[23] Soman, Rohan, Junghyun Wee, and Kara Peters. "Optical fiber sensors for ultrasonic structural health monitoring: A review." Sensors 21, no. 21 (2021): 7345.

[24] Chang, S. Grace, Bin Yu, and Martin Vetterli. "Adaptive wavelet thresholding for image denoising and compression." IEEE transactions on image processing 9, no. 9 (2000): 1532-1546.

[25] Oppenheim, Alan V., and Ronald W. Schafer. 2010. Discrete-Time Signal Processing. 3rd ed. Upper Saddle River, NJ: Prentice Hall.

[26] ASTM C215-14 Standard Test Method for Fundamental Transverse, Longitudinal, and Torsional Resonant Frequencies of Concrete Specimens, ASTM International, West Conshohocken, PA, (2014).

[27] ACI Committee. "Building code requirements for structural concrete (ACI 318-19) and commentary (ACI 318R-19)." American Concrete Institute, (2019).

[28] Lydon, F. D., and R. V. Balendran. "Some observations on elastic properties of plain concrete." *Cement and Concrete Research* 16, no. 3 (1986): 314-324.

[29] Videla, C., Domingo J. Carreira, and N. Garner. "Guide for modeling and calculating shrinkage and creep in hardened concrete." *ACI report* 209 (2008): 76.

[30] Mac-Eteli, H. D., and S. Sopakirite. "An Analytical Review on External and Internal Concrete Curing Mechanisms." *International Research Journal of Engineering and Technology (IRJET)* 8, no. 12 (2021): 1045-1054.

[31] Bahraq, Ashraf A., Mohammed A. Al-Osta, Omar S. Baghabra Al-Amoudi, I. B. Obot, Mohammed Maslehuddin, Habib-ur-Rehman Ahmed, and Tawfik A. Saleh. "Molecular simulation of cement-based materials and their properties." *Engineering* 15 (2022): 165-178.

[32] ASTM International. *Standard Test Method for Flexural Strength of Concrete (Using Simple Beam with Third-Point Loading).* ASTM C78/C78M-18. West Conshohocken, PA: ASTM International, 2019. https://doi.org/10.1520/C0078_C0078M-18

On the EVT-Based Nonlinear Impulsive Noise Suppression Over Indoor Low-Voltage Networks

Steven O. Awino
Dept. of Electronic and Computer Engineering
Durban University of Technology
Durban, South Africa
stevena@dut.ac.za

Bakhe Nleya, Mlungisi Molefe
Dept. of Electronic and Computer Engineering
Durban University of Technology
Durban, South Africa
bakhen@dut.ac.za, mlungisiM3@dut.ac.za

***Abstract*—Impulsive noise remains one of the most critical limiting factors in power line communication (PLC) systems, especially in indoor low-voltage networks where the electromagnetic environment is highly unpredictable. Despite the increasing deployment of PLC technologies for smart homes, IoT devices, and energy management systems, reliable data transmission is still hindered by the extreme, bursty, and non-Gaussian characteristics of impulsive noise. This motivates the need for noise mitigation schemes that are not only adaptive but also statistically grounded in modeling rare and severe disturbances. This paper presents a nonlinear impulsive noise suppression technique over indoor low-voltage networks, based on the Extreme Value Theory (EVT) approach. Impulsive noise present in the indoor low-voltage networks severely degrades data transmission, thus affecting the reliability of communication in this network due to its bursty and non-Gaussian nature. In addition, due to the varying intensity of impulsive noise in the power line network, the traditional filtering approaches that include linear filtering, blanking and code-based approaches fail to provide robust suppression. The approach used herein leverages EVT for the estimation of the threshold levels combined with nonlinear denoising. The Peak Over Threshold (POT) approach is employed to handle the threshold exceedance probability, which is further used for adaptive threshold selection for nonlinear filtering. The results show that this proposed EVT-based method provides a more balanced and statistically reliable suppression in terms of better Root Mean Square Peak Reduction Ratio $(RMSR)$. This demonstrates that the EVT-based nonlinear impulsive noise suppression approach achieves superior performance in mitigating against the extreme and bursty impulsive noise compared to the traditional techniques.**

***Index Terms*—Power Line Communication (PLC), Extreme value theory (EVT), Nonlinear suppression, Low-voltage Networks.**

I. Introduction

The low-voltage indoor electric network, once designed purely to transmit electrical energy, has currently evolved into sprawling data highways transmitting data besides power. This dual use, also called Power Line Communication (PLC), has transformed the ubiquitous low-voltage indoor power network inside homes and buildings into a medium for broadband connectivity without the need for extra cabling [1]–[3]. The motivation lies in its reach and that PLC has the ability to extend data and multimedia services to every outlet already existing in the walls. However, this indoor electric network that faithfully transmits electrical power also serves as a channel for bursty and impulsive noise besides internal and external electromagnetic interferences. These bursty and impulsive noise, generated by the switching of connected electrical appliances, occur unpredictably, overwhelming traditional filters and distorting transmitted data. Their bursty and non-Gaussian nature renders them catastrophic compared to the background noise, motivating the continued search for effective and efficient impulsive noise suppression techniques that are not only mathematically efficient and tractable but also adaptive to the statistical extremes of the low-voltage indoor environment.

Previously, various approaches have been employed to mitigate against impulsive noise in the PLC system. These past approaches relied on blanking and clipping preprocessors [4], [5] that null or clip noise samples that exceeded a fixed amplitude threshold. As such, these approaches are straightforward and mathematically efficient since their performance depended on the careful threshold tuning. However, such a fixed threshold leads to significant distortion in the transmitted signal and residual noise because the fixed threshold may either suppress useful signal peaks or fail to remove strong impulsive noise bursts. In order to mitigate against blanking errors, selective mapping and tone reservation on the transmitter-side preprocessing were proposed in [6]. These techniques lowered the peak-to-average power ratio at the transmitter by reducing the probability of mistaking the useful peak impulses. However, they increase transmitter complexity, decrease spectral efficiency, and cannot be used in scenarios that only allow receiver-side mitigation of problems like multipath interference and noise. In [7], an Adaptive Nonlinear Differential Limiter (ANDL) is proposed as an analogue front-end filter that adaptively switches between linear and nonlinear operation based on a resolution parameter to suppress impulsive noise. Even though this approach outperforms blanking and linear filtering methods by effectively reducing the bit error rates in impulsive noise environments, it is highly sensitive to correct tuning of the resolution parameter. In addition, design complexities and cost are introduced due to the analogue implementation. Consequently, oversampling is required to imitate analogue behaviour accurately, and benefits diminish under moderate noise conditions. As such, ANDL is less flexible compared to EVT-based nonlinear thresholding,

in which the thresholds are statistically derived from extreme noise behaviour, thus allowing adaptiveness, efficiency in computation and enhanced robustness in impulsive noise suppression across varying impulsive noise environments. In [8], a sparse Bayesian learning approach that models impulsive noise as a sparse vector and iteratively approximates and deducts it using null/pilot tones, full-tone processing, and decision feedback is proposed. Even though this method improves the bit error rate and signal-to-noise ratio (SNR) significantly in an OFDM-based PLC system under both asynchronous and cyclostationary impulsive noise, it has high computational complexities, dependent on large observation windows and memory, sensitivity to sparsity, and increased latency due to iterative processing. This is unlike the proposed EVT-based nonlinear thresholding, which is expected to derive adaptive thresholds directly from the extreme noise behaviour. A robust receiver architecture tailored for wavelet-OFDM PLC systems is proposed to mitigate impulsive noise is proposed in [9]. This approach combines a pre-impulsive noise mitigation block, an adaptive discrete wavelet transform per subchannel equalizer, and an iterative impulsive noise reconstruction stage. As such, impairments in channel estimation and equalization introduced by impulsive noise are reduced, and the results show performance approaching that of an ideal no-impulsive noise scenario. However, this approach requires multiple adaptive components, thus increasing computational complexities and resource utilization. In addition, the parameter and threshold levels must be tuned to match the PLC network impulsive noise statistics, which are subject to variation in real deployment; hence, there is a slower reaction to the abrupt changes in noise bursts.

Recently, machine learning techniques such as deep neural networks aided detection in [10] have been applied to solve the problem of impulsive noise suppression. In this approach, models are trained to distinguish between noise and useful signal, thus achieving better results in a controlled setting. Nonetheless, such models require large datasets for training, are mathematically intensive and have a tendency to fail to generalize in the actual time-varying PLC environments. Similarly, statistical models of PLC impulsive noise have been explored, such as in [11], where bursty impulsive noise was modelled using heavy-tailed distributions. Even though such approaches provide meaningful insight into noise characterization, they fail to provide a noise suppression mechanism directly. As such, a much more promising direction lies in the proposed approach herein, Extreme Value Theory (EVT), where the behaviour of extreme outliers in a signal is statistically modelled. The EVT-based nonlinear suppression approach proposed herein possesses adaptive thresholds that are data-driven and tuned by tail modelling and provides a targeted noise suppression of the rare extreme impulsive and bursty noise events, besides being suitable for embedded PLC systems due to their lightweight implementation that does not require training data.

The rest of the paper is organized as follows: In Section II, the proposed theoretical framework is developed for impulsive PLC noise in the frequency range of $1-30$ MHz. Data collection and pre-processing of acquired PLC noise measurements is then presented in Section III. Results and discussion is then presented in Section IV. Finally, the paper ends with concluding remarks in Section V.

II. Theoretical Framework

A. Proposed Signal Model

Assume the received PLC signal at the receiver side is defined by,

$$r(t) = z(t) + n_{bg}(t) + n_{imp}(t) \tag{1}$$

where, $z(t)$ is the desired signal, $n_{bg}(t)$ defines the background noise and $n_{imp}(t)$ is the bursty impulsive noise present in the network. This bursty impulsive noise is characterized with heavy-tailed behaviour that traditional noise filtering methods frequently fail to address.

B. EVT-based Thresholding

Since the interest is in the noise-amplitudes that surpass a threshold, the Peak over threshold (POT) approach is considered in EVT. Let $S_1, S_2, S_3, \ldots, S_N$ be the acquired noise samples. For a suitably high threshold ϑ, the exceedances can be described by [12]–[15],

$$R = S - \vartheta \quad | \quad S > \vartheta. \tag{2}$$

These exceedances follow the Genaralized Pareto distribution (GPD) defined by [12]–[15],

$$F(r) = 1 - \left(1 + \frac{\xi(r-\vartheta)}{\varsigma}\right)^{\frac{-1}{\xi}}, \quad r > \vartheta \tag{3}$$

where, ϑ is the threshold that is set high enough to capture extreme noise events, ξ, is the shape parameter and ς is the scale parameter that determines the variability over the set threshold selected using mean excess Plot.

The adaptive threshold can then be estimated by, [12]–[15],

$$\Gamma = \vartheta + \frac{\varsigma}{\xi}\left[\left(\lambda\tau\right)^{\xi} - 1\right] \tag{4}$$

where, λ defines the exceedance rate that ensures that the threshold adapts to the severity of impulsive noise in real time over an average time τ, and the other parameters bear their usual meaning.

C. Proposed Nonlinear Suppression Framework

The proposed noise suppression herein is composed of the following stages:

i. Detection of exceedance: Detected samples exceeding Γ are flagged as extreme noise events or impulsive.
ii. Nonlinear thresholding: In this stage, the coefficient values above the exceedance threshold Γ are shrunk or nullified.
iii. Median filtering: The residual peaks are suppressed by a slidingwindow median operator. This is to ensure robustness against remaining peaks

III. Data Collection and Processing

To capture the bursty and impulsive characteristics of PLC noise, the measurement setup was designed around the principle of listening to the indoor low-voltage network. A Rigol DS2202A digital storage oscilloscope (DSO) was not only used as a listening device for noise in the power network but also for recording these noise characteristics in the frequency range of 1–30 MHz. To ensure that the dominant 50 Hz carrier in the power network was rejected, a high-pass band coupling circuit was connected between the DSO and the power network as shown in Fig. 1.

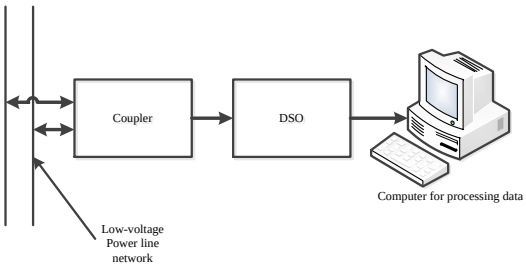


Fig. 1: Experimental set-up for PLC noise measurements.

The DSO did not only ensure continuous streaming of the noise signal to the computer through the USB interface but also enabled the raw measured dataset to be logged in real time. This arrangement allowed samples at 1 Giga-sample per second, resulting in a 0.14-second window length (14 mains cycles). Measurement was carried out from an isolated apartment which is a typical indoor low-voltage power line network that mirrors the diversity of a modern household. This venue contained light dimmers, fluorescent lights, a cathode ray tube television set, a washing machine, two refrigerators, an iron box, a vacuum cleaner, a thermostat, an electric kettle, an electric water heater, an electric cooker, juice blenders, a microwave oven and security lights. This ensured that the collected measurement data reflected the unpredictable reality of PLC noise. A sample of the measurement data is shown in Fig. 2, where impulsive noise events, bursts and background noise can be seen from a visual inspection.

IV. Results and Discussion

A. Noise Suppression Results

In this paper, POT is employed to characterize impulsive noise bursts, while nonlinear thresholding and median filtering is used to suppress them in real-time. Adaptive threshold plot with 95% confidence bound is shown in Fig. 3 where the x-axis is the repeat period, and the y-axis is the repeat level that shows the peak amplitude that is expected to be exceeded once every Γ observations. The red solid line is the predicted repeat level, while the confidence bounds are shown in the green and black dashed lines. For a given repeat period, the peak amplitude that exceeds a specific threshold once every Γ

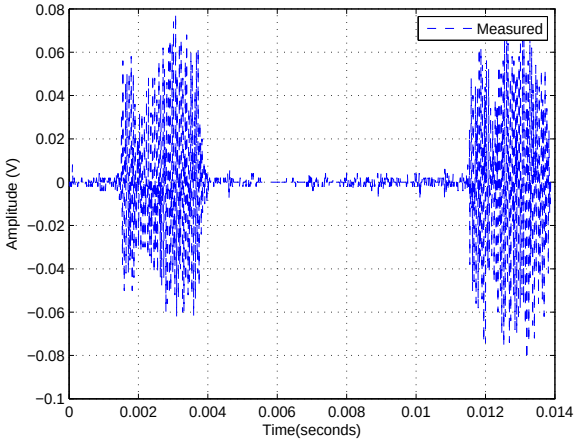


Fig. 2: A sample acquired noise with two impulses

exceedance is estimated. For example, a repeat level of 0.0056 for $\Gamma = 20$ means that a noise burst > 0.0056 is expected every 20 threshold-exceeding impulsive bursts.

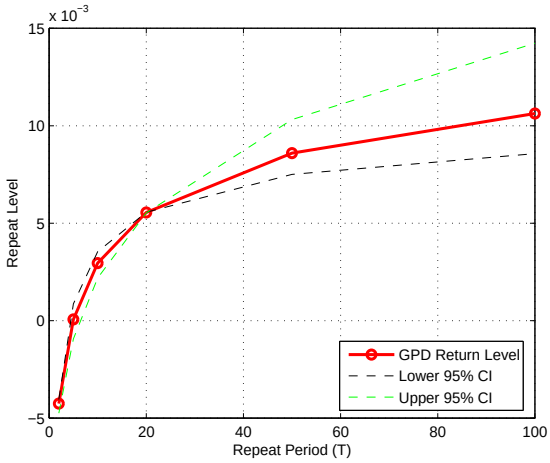


Fig. 3: GPD repeat level plot with 95% confidence interval

From Fig. 4, the excess peaks that are above the threshold value, ϑ can be seen. In addition, this figure shows a comparison of the suppressed signal after EVT-based nonlinear filtering with the raw measured signal. The raw measured signal exhibits high-amplitude and bursty impulses with peaks exceeding five times the root mean square (RMS) of the background and cause catastrophic effects on transmitted data since they mask the modulated carrier thus reducing the accuracy of the demodulation process at the receiver side.

After suppression, the impulsive peaks were reduced by approximately over 100% in amplitude, while the background remain largely intact. Actually the suppressed noise peaks hug closer to the background noise since the suppression spreads the distribution away from the heavy-tailed extremes. This is unlike median-only filtering that tends to distort sharp symbol

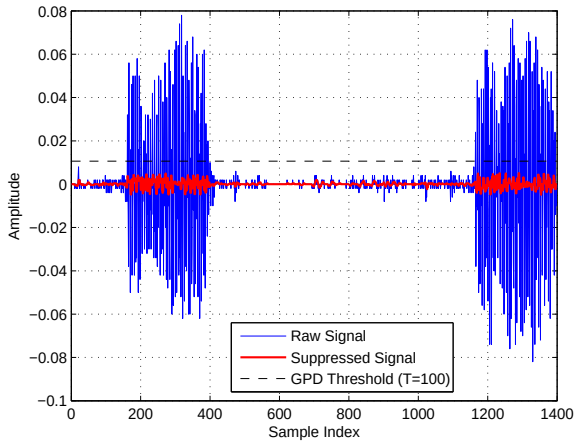


Fig. 4: Suppressed Vs raw noise PLC noise signal

transitions unlike this proposed EVT-thresholding method that removes only statistical extreme outliers selectively. Consequently, this process results in a smoother signal waveform as it also preserves the fine-grain structures which are critical for demodulation.

B. Performance Analysis

To provide a multi-layered measure of effectiveness of the proposed noise suppression approach, these four metrics were used:

1) Peak Reduction Ratio (PRR)*:* The PRR metric provides a means of quantifying the effectiveness of a suppression algorithm in reducing the high-amplitude spikes present in the PLC noise. In addition, it is a critical metric for reducing peak-induced distortion in PLC networks. Mathematically, PRR can be defined as,

$$PRR = 10\log_{10}\left(\frac{P_{peak,raw}}{P_{peak,supp}}\right)\text{dB} \tag{5}$$

where, $P_{peak,raw}$ and $P_{peak,supp}$ define the peak power of the raw and suppressed signals respectively.

2) Root Mean Square Peak Reduction Ratio $(RMSR)$*:* This metric combines the concepts of RMS power and PRR. It expands PRR through normalization of the peak reduction to the RMS level of the suppressed signal, thereby resulting in a more balanced metric. This can be mathematically expressed as,

$$RMSR \quad \text{reduction} = \left(\frac{RMS_{raw} - RMS_{supp}}{RMS_{raw}}\right) \times 100\% \tag{6}$$

where RMS_{raw} and RMS_{supp} define the RMS levels of the noise before and after suppression, respectively. The $RMSR$ is useful in evaluating the performance of nonlinear thresholding, median filtering, or EVT-based suppression methods in reducing the strong spikes present in the measured PLC noise relative to the overall power of the signal.

3) Exceedance Rate (ER)*:* This metric is used to determine and establish the frequency of extreme impulses exceeding the set threshold level. It shows the statistical resilience of the PLC system against the existence of impulsive noise in the network. Mathematically, it can be defined by [12]–[15],

$$\lambda = \frac{n}{N} \times 100\% \tag{7}$$

where n and N are the number of samples exceeding Γ and total number of samples respectively.

4) Signal-to-Noise Ratio (SNR)*:* To determine the overall PLC system data transmission reliability, the SNR metric was employed as follows,

$$SNR = 10\log_{10}\left(\frac{P_{ref}}{P_{noise}}\right)\text{dB} \tag{8}$$

where P_{ref} and P_{noise} are the average power of the reference signal and noise after suppression, respectively. In traditional communication systems, the conventional SNR is normally sufficient to provide information on the distortion severity. However, this leads to underestimation of distortion severity in impulsive noise channels since it averages out impulsive bursts. As such, in this work, the SNR is combined with PRR and ER to provide a fuller picture.

Table I presents a summary of the comparative results between the proposed EVT-based nonlinear suppression, median-only and the fixed threshold noise suppression methods based on the four metrics.

TABLE I: Suppression Performance Summary

Method	$PRR(\%)$	$RMSR(\%)$	$ER(\%)$	SNR **(dB)**
EVT-based	8.3	91.45	100	16.26
Median-only	8.3	89.77	N/A	16.28
Fixed Threshold	8.3	89.77	100	16.28

It can be observed that the PRR values are the same and comparable across all the algorithms under comparison. This means that all the algorithms reduced the present noise peaks by the same degree and that none of them have a decisive advantage in controlling the maximum peaks. For the $RMSR$ values, the proposed EVT-based method with a value of 91.45% shows a slightly lower impulsive noise dominance compared to the median-only and fixed threshold methods such that the noise peaks are closer to the average RMS level. This suggests that the proposed EVT-based suppression has more balance in controlling extreme outliers relative to the signal power, hence managing the noise peaks more efficiently and effectively relative to the RMS level. The EVT-based and fixed threshold methods achieved an ER value of 100% while in the median only, it was either not computed or not applicable. This means that both the EVT-based and fixed threshold methods identified all the noise peaks exceeding the set threshold value, thus suppressing all the present strong burst impulses. However, in the median-only method, the ER is irrelevant since this filtering approach does not seem to operate directly on exceedances. The SNR values remained

nearly the same from observation. This means that none of the suppression methods result in distortion and are comparable in terms of signal recovery qualities.

V. Conclusion

The results herein show that while all three suppression methods are effective in peak and SNR improvement, the EVT-based method provides a more balanced and statistically reliable suppression in terms of better $RMSR$, rendering it more suitable for countering the effects of bursty impulsive noise in PLC systems. This is because EVT-derived thresholds adapt to statistical extremes as modelled by the GPD with signal preservation, outperforming median filtering and fixed threshold methods in both quantitative metrics and waveform integrity. As such, this outcome is particularly important for indoor PLC systems, where short burst intervals lead to clustering of errors.

Acknowledgment

This work was supported by Durban University of Technology under Harnessing Talent (HANT) Postdoctoral Fellowship (PDF) Programme. Further, we also acknowledge Prof. Thomas Afullo from University of KwaZulu-Natal for facilitating measurement used in this research.

References

[1] H. Meng, Y. L. Guan, and S. Chen, "Modeling and analysis of noise effects on broadband power line communications," *IEEE Trans. On Power Delivery*, vol. 20, no. 2, pp. 630–637, 4 2005.

[2] R. M. Vines, H. J. Trussell, L. J. Gale, and J. B. O. JR., "Noise on residential power distribution circuits," *IEEE Trans. On Electromagn. Comp.*, vol. EMC-26, 1984.

[3] M. Zimmermann and K. Dostert, "Analysis and modelling of impulsive noise in broad-band power line communications," *IEEE Trans. On Electromagn. Comp.*, vol. 44, no. 1, pp. 249–258, 2 2002.

[4] C. R. Nokes, O. P. Haffenden, J. V. Mitchell, A. P. Robinson, J. H. Stott, and A. Wiewiorka, "Detection and removal of clipping in multicarrier receivers," *European patent application EP1043874*, vol. 4, pp. 785–800, 10 2000.

[5] S. V. Zhidkov, "Analysis and comparison of several simple impulsive noise mitigation schemes for ofdm receivers," *IEEE Transactions on Commun.*, vol. 56, no. 1, pp. 5–9, 2008.

[6] K. M. Rabie and E. Alsusa, "Improving blanking/clipping based impulsive noise mitigation over powerline channels," in *Proceedings of 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 9 2013, pp. 276–280.

[7] R. Barazideh, B. Natarajan, A. V. Nikitin, and R. L. Davidchack, "Performance of analog nonlinear filtering for impulsive noise mitigation in OFDM-based PLC systems," in *Proceedings of 2017 IEEE 9th Latin-American Conference on Communications (LATINCOM)*, 2017, pp. 1–6.

[8] L. Jing, M. Nassar, and B. L. Evans, "Impulsive noise mitigation in powerline communications using sparse bayesian learning," *IEEE Journal on Selected Areas in Commun.*, vol. 31, no. 7, pp. 1172–1183, 6 2013.

[9] Y.-R. Chien and H.-C. Yu, "Mitigating impulsive noise for wavelet-(OFDM) powerline communication," *Energies 2019*, 4 2019.

[10] G. Yang, Y. Qian, Z. Wang, X. Zhou, and W. Wu, "An intelligent impulsive noise mitigation with deep learning method," *International Journal of Mechanical System Dynamics*, vol. 4, no. 3, pp. 346–360, 9 2024.

[11] F. Chelangat and T. J. O. Afullo, "Time series modelling of powerline communication impulsive noise: Queuing theory approach," *Progress In Electromagnetics Research C*, vol. 150, pp. 157–168, 2024.

[12] .-e. Longin François Michel, *Extreme Events in Finance: A Handbook of Extreme Value Theory and its Applications*. New Jersey, USA: John Wiley & Sons, Inc., 2017.

[13] S. Kotz and S. Nadarajah, *Extreme Value Distributions: Theory and Applications*. London: Imperial College Press, 2000.

[14] J. T. Jan Beirlant, Yuri Goegebeur and J. Segers, *Statistics of Extremes: Theory and Applications*. West Sussex, England: John Wiley & Sons, Inc., 2004.

[15] J. Maria, N. Cláudia, and D. V. Greetham, *Extreme Value Theory: Forecasting and Assessing Risk of Individual Electricity Peaks*. Cham: Springer International Publishing, 2020.

Rethinking Graduate Skills Integration for the Fourth Industrial Revolution in South African Universities

Rennie Naidoo
School of Business Sciences
University of the Witwatersrand
Johannesburg, South Africa
rennie.naidoo@wits.ac.za

Naomi Isabirye
line 2: School of Business Sciences
University of the Witwatersrand
Johannesburg, South Africa
naomi.isabirye@wits.ac.za

Olujoruwon Michael Samuel
School of Business Sciences
University of the Witwatersrand
Johannesburg, South Africa
Olorunjuwon.Samuel@wits.ac.za

***Abstract*— Despite significant advances in South Africa's connectivity and computing infrastructure over recent years, these infrastructural gains have not proportionately translated into improvements in innovation, transformation, or global competitiveness. The gap in critical Information and Communication Technology (ICT) skills has been increasingly recognised as a barrier to the country's ability to leverage opportunities associated with the Fourth Industrial Revolution (4IR). This has raised concerns about whether universities in South Africa are adequately preparing graduates with the appropriate skills to participate meaningfully in the digital economy. This paper evaluates the integration of 4IR skills in South African universities, exploring how experts conceptualise the competencies that students need to succeed in a rapidly evolving digital economy. Using a phenomenographic approach, the study investigates variations in how technical, soft, entrepreneurial, and interdisciplinary skills are understood, with implications for curriculum design and workforce preparedness. Drawing on qualitative interviews and focus groups, the research highlights challenges in integrating 4IR skills into education, including institutional disparities, resource limitations, and the undervaluing of interdisciplinary learning. The findings emphasise the need for tailored curriculum reforms that balance technical expertise, industry collaboration, and work-integrated learning (WIL) to align education with industry needs. Additionally, the study underscores the importance of early computational thinking and ICT integration in school curricula to ensure foundational digital literacy. By capturing diverse stakeholder perspectives, this research informs inclusive education strategies that prepare students for South Africa's evolving digital economy.**

Keywords*— Digital Skills Gap, Fourth Industrial Revolution (4IR), Economic Restructuring, Universities, South Africa, Curriculum Design, Industry Collaboration, Phenomenography.*

I. INTRODUCTION

The past decade has seen an expansion in broadband reach and speed, greater mobile penetration, and an overall improvement in connectivity for business and society in South Africa [1]. These advancements have coincided with extensive shifts in the global technological, economic, and social landscape spurred by the Fourth Industrial Revolution (4IR). However, while the infrastructure and technology to support digital transformation have improved, South Africa continues to play a "catch-up game" in 4IR strategy and innovation compared to countries such as the United States, Germany, China, and Kenya [2]. The skills gap remains a major barrier to leveraging 4IR opportunities [2], [3], [4], [5], [6]. Yet this shortage coexists with a notable number of unemployed ICT graduates [7]. This raises the question of whether South African universities are adequately equipping graduates with the ICT skills needed in the workplace [7], [8]. There is an apparent disjunction between the number of ICT graduates, the high demand for critical ICT skills, and the limited absorption of graduates into areas of need. Consequently, the integration of 4IR skills in universities is gaining traction globally, including in South Africa, where universities are under pressure to prepare future-fit graduates [9], [10], [11].

Scholars argue that disciplinary knowledge alone is insufficient for graduates to thrive in 4IR workplaces [8]. Traditional classroom-based learning is no longer adequate to produce graduates capable of navigating complex technological environments [12], [13], [14]. Graduates now require cognitive, behavioural, technical, and ethical skills essential for 4IR contexts [15], [16]. This shift emphasises the need for interdisciplinary and digital skills, such as critical thinking, problem-solving, and Artificial Intelligence (AI) proficiency [9], [16]. With youth unemployment remaining high, universities must align curricula with industry demands to prepare students for rapidly changing technological landscapes [17]. South Africa requires significant contributions from education systems to shape a workforce capable of addressing these challenges [18], [19].

However, substantial obstacles remain. A shortage of managers, researchers, and technically skilled workers limits the country's ability to compete globally and leverage 4IR for economic growth [20], [3]. Rapid technological innovations place many African workers at risk of displacement due to automation [21], [4]. Human development has the potential to positively influence 4IR adoption, yet its impact remains limited without targeted efforts to build relevant skills. Weak governance, policy delays, and inadequate infrastructure further hinder the effective integration of 4IR across the continent [3]. To bridge these gaps, universities must prioritise curriculum reform, industry collaboration, and initiatives that foster innovation, ensuring graduates are prepared for a technologically advanced global economy [22].

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

II. LITERATURE REVIEW

The Fourth Industrial Revolution (4IR) is characterised by the convergence of digital, physical, and biological systems, enabled by advancements in big data, robotics, artificial intelligence (AI), automation, the Internet of Things (IoT), and cloud computing [23]. These technological shifts continue to reshape global workforce needs, requiring graduates to possess a combination of technical and non-technical competencies. The integration of these skills into universities is crucial for preparing students to navigate the rapidly evolving technological landscape [27]. These skills aim to equip graduates with the technical expertise, interdisciplinary capabilities, and adaptability needed to address the complex demands of modern industries [9]. This section reviews existing literature on the diversity of 4IR skill requirements, their implications for curriculum design, and how current teaching practices align with workforce demands.

4IR skills offer students the opportunity to become productive members of the workforce by engaging in real-world technological applications that generate social and economic value. The literature highlights critical competencies such as critical thinking, problem-solving, adaptability, and interdisciplinary collaboration as central to preparing students for the 4IR workforce [28], [10]. These competencies reflect the varied ways in which academics and industry experts conceptualise 4IR skills, emphasising both technical and soft skills as essential for work readiness.

Studies further suggest that exposure to technology-driven projects helps students bridge the gap between theoretical knowledge and practical application. This development fosters critical competencies in emerging fields like artificial intelligence (AI), data analysis, and robotics, enabling graduates to adapt to the dynamic job market[27], [28]. Such findings support the need to further understand the diverse perspectives on skill requirements and their implications for curriculum innovation.

The integration of 4IR skills into curricula often depends on collaboration with industry partners, which ensures that students gain practical exposure to emerging technologies [28], [29] These collaborations allow students to engage in hands-on learning with cutting-edge tools and techniques relevant to their future careers. Direct interaction with industry professionals not only enhances students' technical proficiency but also fosters critical employability skills, such as teamwork and communication, aligning educational outcomes with workforce needs [30].

Industry partnerships also help bridge resource gaps by providing access to advanced technologies and funding for research, benefiting both students and universities [28]. Research also consistently emphasizes the positive impact of 4IR skills on career prospects. Graduates proficient in AI, machine learning, and related technologies are highly sought after in industries such as technology, finance, and manufacturing, where technical expertise drives innovation and efficiency [29]. While some concerns exist about overemphasizing vocational skills at the expense of broader academic development, the literature indicates that integrating 4IR skills ensures a balance between academic rigor and professional readiness [28], [31].

A growing body of research connects 4IR competencies to entrepreneurial success, highlighting the need for universities to foster not only employability but also innovation and self-employment [32]. Graduates with both technical and interpersonal skills are more likely to launch successful businesses, contributing to job creation and economic growth [33], [34]. These findings emphasize the role of universities in equipping graduates with diverse skills that extend beyond traditional employment pathways.

Building on this premise, this article investigates whether universities effectively prepare graduates with competencies needed in the 4IR. By employing a phenomenographic methodology [35], the research explores variations in academic and industry perspectives and provides actionable insights into how curriculum design can be reimagined to address both the challenges and opportunities of the 4IR era.

III. RESEARCH METHODOLOGY

This study employs a phenomenographic research approach to explore the diverse perspectives of academics and industry leaders on the skills required for graduates to succeed during 4IR. Phenomenography aims to capture the qualitative variation in how individuals experience and conceptualise specific phenomena [36]. Rather than generalising, the approach seeks to reveal the different ways in which participants understand the phenomenon—in this case, the graduate skills needed for 4IR in South African universities.

Phenomenography is particularly suitable for this research because it aligns with the constructivist epistemology underpinning the study of 4IR [35]. It emphasises the subjective interpretation of reality and focuses on understanding diverse experiences and conceptions[37], [38]. This approach allows for a deeper exploration of expert perspectives [39], ensuring that recommendations for curriculum design address the varied needs and contexts of South African universities.

The research was conducted with a diverse group of 10 experts from multiple South African universities and industries. These participants included academics specialising in computing, mathematical sciences, and 4IR technologies, as well as economic and innovation researchers, human sciences researchers, and industry leaders. Their perspectives provided valuable insights into the integration of emerging technologies, curriculum development, and strategies for enhancing South Africa's readiness for 4IR. Additionally, participants offered reflections on the societal and economic impacts of 4IR and practical approaches to addressing industry requirements.

Data collection and analysis

Data collection involved semi-structured interviews with the 10 participants, each lasting between 45 and 80 minutes, with an average duration of one hour. The open-ended interview format encouraged participants to reflect on their personal experiences and views regarding the teaching of 4IR skills, the challenges of curriculum integration, and their perspectives on preparing graduates for the 4IR workforce. Interview questions also addressed strategic initiatives in universities to ensure industry

readiness, providing rich qualitative data on how participants conceptualise and experience the integration of 4IR skills.

The interviews were transcribed verbatim, and the analysis followed a phenomenographic tradition, focusing on identifying descriptive categories that reflect the different ways in which participants understand 4IR skills. This second-order perspective meant that our analysis concentrated not on the objective characteristics of 4IR skills but on how participants conceptualise and experience these skills in their specific contexts. The process involved thematic analyses [40], which focused on identifying, analysing, and reporting patterns in participants' conceptions of 4IR skills.

These analyses were used to develop an outcome space, which illustrates the spectrum of perspectives held by participants. For example, some experts emphasised the importance of technical proficiency in areas such as software coding, AI, and data analysis, while others highlighted the significance of interdisciplinary skills such as critical thinking, adaptability, and collaboration.

By emphasising the relationship between participants and the phenomenon, phenomenography ensures that findings are contextually bound and dependent on this interaction [36], [37] For instance, experts from resource-rich institutions described advanced strategies for integrating AI and machine learning. At the same time, those from resource-constrained universities emphasised the need for foundational digital literacy and equitable access to technology. This relational focus provides nuanced insights into how institutional capacity shapes 4IR skill integration.

Ethical considerations were strictly maintained throughout the study. All participants provided informed consent before participating in the interviews, and the research adhered to ethical guidelines to protect the participants' privacy. Confidentiality was ensured by anonymising all personal information.

IV. RESULTS

The expert interviews provided rich insights into the conceptions of integrating 4IR skills into universities from a diverse group of academics and industry leaders. The analysis of these interviews revealed various ways in which experts understand and experience the challenges and opportunities posed by 4IR for curriculum development, institutional readiness, and student employability.

Conceptions of Technological Trends and Their Impact on Universities

A dominant conception among participants revolves around the urgent need for universities to adapt to the rapid technological changes brought about by 4IR. Many experts noted a significant skills gap, particularly in the areas of coding, AI, and software development, which are now essential for a wide range of industries. One industry expert expressed the following concern:

"I think the biggest problem in corporate South Africa is lack of skills. Large organisations need people with skills in coding for operational efficiency. With the digitisation of the work system, every employee wants new features on their apps, and this requires a coder."

This conception highlights the importance of integrating technical skills into the curriculum at earlier stages, particularly in undergraduate programs. Universities are seen as needing to prepare students for immediate challenges, equipping them with foundational digital skills to remain competitive in the workforce. Additionally, participants pointed to the transformation of traditional industries through digital platforms, particularly in sectors like the creative industry. An industry expert remarked:

"4IR technologies have significant implications for the creative industry on so many levels which is the sector that I focus on most. In the creative industries, distribution no longer happens via distribution company it's on a digital platform and I can download my music directly. So, it's not just in the creative industries, but in general the economy structure is shifting to this decentralised network."

This transformation reinforces the conception that adaptability and innovation are essential in preparing graduates, especially as industries evolve in response to new technologies. universities should prioritise curricula that foster both digital fluency and creative problem-solving. Preparing students for these changing roles ensures they can participate in decentralised networks and lead the innovative processes that will shape the future of industries.

Conceptions of Industry Collaboration and Work-Integrated Learning

A recurring understanding across interviews was the importance of industry collaboration in bridging the gap between academic learning and practical experience. Experts frequently described how partnerships between universities and industries offer invaluable opportunities for students to apply theoretical knowledge in real-world contexts. One participant noted:

"We want to develop our industrial engineering incubator to become an ecosystem where we will be able to tap into other technologies in solving problems. For example, if we've got a cybersecurity laboratory infrastructure, if we've got a virtual reality infrastructure, then experts can just walk across the cybersecurity lab and get solutions to several security problems. With this, the university can partner with government in solving security challenges in the country."

This conception emphasises the importance of practical, hands-on experience in complementing traditional academic approaches. Universities are increasingly seen as needing to engage directly with industry to provide students with the necessary tools and experiences to succeed in the 4IR job market. The collaboration between academia and industry also provides access to cutting-edge technologies and funding for research, which many universities are not able to afford independently. Some universities lack the connections or infrastructure (e.g., cybersecurity labs, VR tools) needed to facilitate partnerships and offer students hands-on experience with cutting-edge technologies.

In addition to industry partnerships, many experts stressed the value of work-integrated learning, where students can gain hands-on experience through internships, co-operative programs, or collaborative projects with industry professionals. An academic noted:

"I have a funded collaborative project in a petrochemical company where some of my students have opportunity to learn on the job and one of them spent almost one year working with the engineers and material science group to gain some experience on the application of machine learning and understanding on the density of states of materials and all other things."

This conception highlights the importance of exposing students to real-world challenges and integrating them into the workforce before graduation. Such industry collaborations provide students with hands-on learning opportunities that bridge the gap between theoretical knowledge and practical application, allowing them to engage with cutting-edge technologies. Moreover, these experiences foster critical problem-solving skills, enhance their ability to work in interdisciplinary teams, and build professional networks that can be crucial for career advancement. However, experts from institutions with fewer connections to corporations face difficulties in providing students with work-integrated learning opportunities, emphasising disparities in resources and access.

Conceptions of Gaps in 4IR Curriculum

A significant concern raised during the interviews was the current state of 4IR curricula in South African universities. Experts frequently pointed out that while there is growing recognition of the need for 4IR skills, there remain considerable gaps in how these technologies are integrated into degree programs. One participant highlighted:

"I would actually like a scenario where we have basic 4IR modules offered across academic disciplines maybe at first year level. However, students don't take courses that are not compulsory seriously, especially non-core courses."

This conception calls for comprehensive curriculum reform, where technologies such as AI, machine learning, and virtual reality are incorporated into undergraduate education. The experts argued that the pace of technological advancement makes it crucial for students to gain exposure to these skills early in their academic journeys, ensuring that graduates are equipped to handle the challenges of the 4IR from the moment they enter the workforce. Moreover, embedding these skills into core curricula ensures that all students, regardless of their primary field of study, are equipped with the digital fluency and critical competencies needed to participate in a technology-driven workforce. This approach aligns with the growing interdisciplinarity of 4IR, where skills such as data analysis, coding, and systems thinking are increasingly relevant across professions, from healthcare and education to engineering and the creative arts. However, some institutions have not yet embedded 4IR skills into compulsory curricula, which reflects variations in how seriously these skills are integrated across programs.

Experts also contended that simply offering these modules as electives may not suffice, as students often deprioritise non-core courses. Instead, integrating these skills into compulsory courses or as interdisciplinary projects can encourage student engagement while fostering a deeper understanding of their practical applications. For instance, a business student learning about AI-driven market analysis or a sociology student exploring virtual reality for social research demonstrates how 4IR skills can enrich diverse academic pathways.

Furthermore, experts expressed the need for curricula to balance technical skills with soft skills such as creativity, critical thinking, and communication. As one academic explained:

"I actually think that students need to have both technical and soft skills. Our programme has been technical because it started as a Diploma in Information Technology and Application Development. So, while the focus was on application development, we have offered modules in communication skills, creative design, ethics, team building, project management, etcetera."

The conception here reflects the growing understanding that well-rounded graduates, who possess both technical proficiency and interpersonal skills, are better prepared for the complexities of the modern workforce. Modules in communication skills, creative design, ethics, and project management equip students not only to execute tasks but also to lead teams and make ethical decisions. However, some institutions tend to focus heavily on technical skills or traditional academic areas, while others adopt a more interdisciplinary approach. The varying emphases on technical, soft, and entrepreneurial skills reflect differences in institutional priorities.

Conceptions of Entrepreneurial Skills and Employability

Finally, a strong conception among the experts was the growing expectation that universities should not only produce employable graduates but also entrepreneurial individuals capable of creating their own opportunities. Several participants noted examples of graduates who had gone on to launch their own businesses using 4IR technologies. One expert shared:

"I just had two students who graduated with cum laude on a visual reality project. Not only did they graduate, but the other one has set up a company called 'X Reality Labs' that actually develops engineering content for universities and industries."

This conception underscores the importance of entrepreneurial education, which equips students with the skills to innovate and create jobs, rather than simply seeking employment. The experts saw this entrepreneurial drive as crucial in an economy increasingly shaped by digital technologies and self-employment opportunities. The concept of graduateness—where graduates are not just ready to join the workforce but are also capable of contributing meaningfully to society—was a recurring theme. As one participant put it:

"I'm just thinking …in the machine learning data driven techniques I will say that you can start your own consultancy because I see some of our senior students who work for the industry for a while and then establish their own consultancy. There are a lot of opportunities like you can work as a quantitative analyst, risk analyst and consult for banks and insurance industry."

This conception reflects a broader understanding of employability in the 4IR era, where success is defined by a combination of technical skills, entrepreneurial spirit, and the ability to adapt to changing environments. However, institutions with fewer resources struggle to provide the same level of entrepreneurial support, such as incubators, mentorship programs, or access to industry networks, which can limit opportunities for students to develop and launch their ideas. The emphasis on entrepreneurial thinking also varied across experts depending on their institutions, with some focusing more heavily on technical or employability skills.

V. DISCUSSION

This section presents the phenomenographic interpretation of expert interviews, uncovering the qualitatively different ways in which participants perceive the digital skills gap and 4IR-driven economic restructuring in South Africa. The findings illustrate variations in how experts conceptualise the integration of 4IR skills into universities, reflecting divergent views on the balance between technical expertise, interdisciplinary competencies, entrepreneurial skill development, and the impact on student career prospects.

Bridging Skills and Employability

Bridging the gap between graduate skills and employability requires concerted, institution driven measures that are sensitive to the unique contexts of the respective institutions. In addition, graduate skills must respond to the dynamic interface between technology, business and people, and emphasise attributes such as innovation, adaptability, and cross-functional collaboration.

The development of technical competencies such as coding, AI literacy, data analytics, and computational thinking emerged as an important measure for addressing the skills needs of the labour market. However, the results also emphasise the weight of soft skills and the need to balance between digital, cognitive, and interpersonal capabilities to navigate complex modern workplaces.

The results also revealed gaps in how 4IR competencies are incorporated into South African university curricula, outlining the need for a more coordinated and intentional institutional stance towards developing these critical skillsets. The findings assert the need for compulsory, foundational 4IR modules across disciplines to mitigate uneven exposure to critical digital skills.

The historical constraints emanating largely from resource disparities and differentiated strategic priorities call for contextualised responses rather than uniform implementation.

Contextual Variations in Industry Collaboration

Industry collaboration emerged as a crucial element affecting graduate skills. Collaboration with industry partners was recognised as essential for providing students with hands-on learning opportunities, particularly through Work Integrated Learning (WIL). WIL creates opportunities for students to reconcile theory and practice while providing access to cutting-edge technologies, enhancing their practical readiness, and building critical problem-solving skills. However, for universities with limited access to industry partners, the absence of such opportunities exacerbates inequities in 4IR readiness across institutions.

These findings echo existing literature on the significance of WIL in bridging the gap between academic learning and workforce demands [29], [31]. However, this study adds depth by identifying systemic barriers that prevent under-resourced universities from establishing and sustaining meaningful industry collaborations. Addressing these disparities will require policy interventions and funding support to ensure that all institutions can provide students with equitable access to industry-driven learning opportunities.

Employability and Entrepreneurship in the 4IR Landscape

Traditionally, gaining technical expertise has occurred within large corporations, universities, and structured industries. However, findings indicate a shift where students are bypassing these institutions—starting their own consultancies, self-distributing creative work, and forming digital-native enterprises. Experts conveyed a range of perspectives regarding the impact of 4IR skills on student career prospects. For many, acquiring 4IR skills was seen as directly linked to enhanced employability, enabling students to navigate complex technological environments and meeting industry demands. Although participants emphasised that 4IR education equips graduates with a competitive edge in the job market, there was also an emerging appreciation that 4IR skills not only improve employability but also foster entrepreneurial opportunities. Some participants cited examples of students leveraging their technical training to launch successful businesses, such as virtual reality content companies and machine-learning consultancies. This finding illustrates the transformative potential of 4IR education in enabling graduates not only to adapt to existing job roles but also to create new market opportunities through innovation. These findings suggest that universities should incorporate modules that encourage entrepreneurial thinking alongside technical training [34]. This also aligns with the greater imperative of the South African Government to promote entrepreneurship through small business development, as a vehicle for economic growth.

The role of universities is shifting from traditional educational models toward functioning as incubators of innovation. No longer confined to teaching and research, universities are increasingly facilitating entrepreneurship and industry partnerships. A university experience is extending beyond theoretical instruction, emphasising real-world application and interdisciplinary problem-solving. This transformation signals a broader redistribution of economic power, where knowledge and opportunity are no longer monopolised by centralised institutions but are instead diffused across networked, decentralised, and entrepreneurial models. The rise of independent ventures and digital platforms suggests that 4IR is not solely about acquiring new technical skills but also about leveraging digital tools to establish alternative career pathways beyond conventional employment structures.

Insights from Phenomenographic Analysis

This study demonstrates the value of phenomenography in capturing nuanced perspectives on the same phenomenon [35], [37]. It moves beyond generalised accounts to identify the unique complexities of 4IR skill integration in South African universities. The findings reveal not only the diverse ways in which participants perceive the benefits of 4IR education but

also the systemic barriers that limit its implementation. By emphasising the contextual challenges and opportunities associated with 4IR skill development, this research extends the existing discourse on curriculum reform, policy-making, and industry collaboration. Ultimately, the study highlights the need for an inclusive and adaptive approach to integrating 4IR skills into universities.

VI. LIMITATIONS

This study employed a phenomenographic approach to capture the variation in expert conceptions of the skills required for the Fourth Industrial Revolution (4IR) in South African higher education. While the methodology is well suited to exploring differences in experience and understanding, the participant sample consisted of only ten experts. Although the diversity of the sample enabled the study to capture a broad spectrum of viewpoints, the small number of participants and wide spread of disciplinary backgrounds may have influenced the depth of convergence within any single field. A more discipline-specific sample—such as experts drawn exclusively from computing, engineering, or data science may have generated more granular and concrete recommendations for curriculum reform within those specific domains.

VII. CONCLUSION

The findings of this study reveal a diverse range of perspectives on integrating 4IR skills into universities. Through a phenomenographic approach, it became evident that while there is a consensus on the critical need for 4IR technologies—such as artificial intelligence and machine learning—to be integrated into university curricula, the experiences of implementing these technologies vary significantly based on resource availability and institutional capacity. Experts emphasised the role of industry collaboration as essential in providing students with practical experience, though challenges in maintaining such partnerships were also noted, particularly for resource-constrained institutions. Additionally, the study highlighted the importance of 4IR education in enhancing student employability and fostering entrepreneurial skills, with some participants viewing these skills as crucial for self-employment and innovation. By considering the diverse ways industry practitioners and academics conceptualise 4IR skills, curriculum design can be more effectively tailored to equip graduates with the competencies required to thrive in an increasingly digital world of work.

VIII. REFERENCES

[1] Independent Communications Authority of South Africa, "The State of the ICT Sector Report of South Africa: Overview of the ICT Sector," Mar. 2025.

[2] T. Marwala, Closing the gap : The Fourth Industrial Revolution in Africa. Pan Macmillan, 2022.

[3] E. Sutherland, "The Fourth Industrial Revolution – The Case of South Africa," Politikon, vol. 47, no. 2, pp. 233–252, Apr. 2020, doi: 10.1080/02589346.2019.1696003.

[4] A. O. Alabi and S. M. Mutula, "Human development for the fourth industrial revolution: Which way for Sub-Saharan Africa?," Dev South Afr, vol. 39, no. 4, pp. 528–542, Jul. 2022, doi: 10.1080/0376835X.2022.2098090.

[5] N. Carrim, "4IR in South Africa and some of its educational implications," Journal of Education, no. 86, pp. 1–18, Apr. 2022, doi: 10.17159/2520-9868/i86a01.

[6] M. Khuluvhe and H. Bhorat, Skills Supply and Demand in South Africa-Labour Market Intelligence Programme. Department of Universities and Training, 2022. [Online]. Available: www.dhet.gov.za

[7] K. Ohei and R. Brink, "Investigating the prevailing issues surrounding ICT graduate employability in South Africa: A case study of a South African university," The Independent Journal of Teaching and Learning, vol. 14, no. 2, pp. 29–42, Sep. 2019.

[8] A. Al-Maskari, T. Al Riyami, and S. Ghnimi, "Factors affecting students' preparedness for the fourth industrial revolution in universities institutions," Journal of Applied Research in Universities, vol. 16, no. 1, pp. 246–264, Jan. 2024, doi: 10.1108/JARHE-05-2022-0169.

[9] T. S. Moloi and M. K. Salawu, "Institutionalizing Technologies in South African Universities towards the Fourth Industrial Revolution," International Journal of Emerging Technologies in Learning (iJET), vol. 17, no. 03, pp. 204–227, Feb. 2022, doi: 10.3991/ijet.v17i03.25631.

[10] N. Sehlako, M. I. Chibambo, and J. J. Divala, "The Fourth Industrial Revolution in South Africa's basic education: a search for cogent curriculum justice," Front Educ (Lausanne), vol. 8, Oct. 2023, doi: 10.3389/feduc.2023.1209511.

[11] D. Mhlanga and T. Moloi, "COVID-19 and the Digital Transformation of Education: What Are We Learning on 4IR in South Africa?," Educ Sci (Basel), vol. 10, no. 7, p. 180, Jul. 2020, doi: 10.3390/educsci10070180.

[12] S. Singaram and C.-H. Mayer, "The influence of the Fourth Industrial Revolution on organisational culture: An empirical investigation," Front Psychol, vol. 13, Nov. 2022, doi: 10.3389/fpsyg.2022.919157.

[13] V. Marivate, P. Aghoghovwia, Y. Ismail, F. Mahomed-Asmail, and S.-L. Steenhuisen, "The Fourth Industrial Revolution – what does it mean to our future faculty?," S Afr J Sci, vol. 117, no. 5/6, May 2021, doi: 10.17159/sajs.2021/10702.

[14] O. S. Aboderin and M. Havenga, "Essential skills and strategies in higher education for the Fourth Industrial Revolution: a systematic literature review," South African Journal of Higher Education, vol. 38, no. 2, pp. 24–43, 2024.

[15] K. Mabe and K. J. Bwalya, "Critical soft skills for information and knowledge management practitioners in the fourth industrial revolution," SA Journal of Information Management, vol. 24, no. 1, Jun. 2022, doi: 10.4102/sajim.v24i1.1519.

[16] W. Maisiri, H. Darwish, and L. van Dyk, "An Investigation of Industry 4.0 Skills Requirements," South African Journal of Industrial Engineering, vol. 30, no. 3, Nov. 2019, doi: 10.7166/30-3-2230.

[17] Presidential Commission on the Fourth Industrial Revolution, "Summary Report and Recommendations presented by Commission on the Fourth Industrial Revolution," 2020. [Online]. Available: www.gpwonline.co.za

[18] A. Fataar, "The emergence of an education policy dispositif in South Africa: An analysis of educational discourses associated with the fourth industrial revolution," Journal of Education, no. 80, 2020, doi: 10.17159/2520-9868/i80a01.

[19] M. Masinde and P. Soux, "Transforming South Africa's Univerities of Technology: A Roadmap Through 4IR lenses," Journal of Construction Project Management and Innovation, vol. 10, no. 2, Dec. 2020, doi: 10.36615/jcpmi.v10i2.405.

[20] S. Lubinga, T. C. Maramura, and T. Masiya, "The Fourth Industrial Revolution Adoption: Challenges in South African Universities Institutions," Journal of Culture and Values in Education, vol. 6, no. 2, pp. 1–17, Mar. 2023, doi: 10.46303/jcve.2023.5.

[21] M. M. Mtotywa, M. A. Seabi, T. J. Manqele, S. P. Ngwenya, and M. Moetsi, "Critical factors for restructuring the education system during the era of the fourth industrial revolution in South Africa," Dev South Afr, vol. 41, no. 1, pp. 16–37, Jan. 2024, doi: 10.1080/0376835X.2023.2234399.

[22] O. S. Madumo and J. R. Kimaro, "Accelerating the Fourth Industrial Revolution in Universities: Realities and Lessons from Universities in Kenya, Zambia and South Africa During the Covid-19 Pandemic," European Journal of Economics, Law and Social Sciences, pp. 196–208, Dec. 2021, [Online]. Available: www.iipccl.org

[23] K. Schwab, The fourth industrial revolution. Portfolio Penguin, 2017.

[24] F. M. Kamaruzaman, R. Hamid, A. A. Mutalib, M. S. Rasul, M. Omar, and M. F. A. M. Zaid, "Exploration and verification of fourth industrial revolution generic skills attributes for entry-level civil engineers," International Journal of Evaluation and Research in Education, vol. 12, no. 1, pp. 121 – 130, 2023, doi: 10.11591/ijere.v12i1.24135.

[25] J. Aliu, D. Aghimien, C. Aigbavboa, A. E. Oke, and A. Ebekozien, "An Employability Skills Model for Built Environment Graduates: A Partial Least Squares Structural Equation Modeling Analysis," Int J Constr Educ Res, vol. 21, no. 1, pp. 74 – 93, 2025, doi: 10.1080/15578771.2024.2333403.

[26] A. Leung, A. Baccaglini-Frank, M. A. Mariotti, and E. Miragliotta, "Enhancing Geometric Skills with Digital Technology: The Case of Dynamic Geometry," Springer International Handbooks of Education, vol. Part F3098, pp. 409 – 437, 2024, doi: 10.1007/978-3-031-45667-1_15.

[27] L. I. Nwosu, M. C. Bereng, T. Segotso, and N. B. Enebe, "Fourth Industrial Revolution Tools to Enhance the Growth and Development of Teaching and Learning in Universities Institutions: A Systematic Literature Review in South Africa," Research in Social Sciences and Technology, vol. 8, no. 1, pp. 51–62, Apr. 2023, doi: 10.46303/ressat.2023.4.

[28] M. Mbiza and S. Sinha, "The Fourth Industrial Revolution: Conceptual paradox or catalyst for achieving the Sustainable Development Goals?," S Afr J Sci, vol. 119, no. 7/8, May 2023, doi: 10.17159/sajs.2023/16090.

[29] W. Maisiri and L. van Dyk, "Industry 4.0 Readiness Assessment for South African Industries," South African Journal of Industrial Engineering, vol. 30, no. 3, Nov. 2019, doi: 10.7166/30-3-2231.

[30] N. J. Van Rensburg, A. Telukdarie, and P. Dhamija, "Society 4.0 applied in Africa: Advancing the social impact of technology," Technol Soc, vol. 59, p. 101125, Nov. 2019, doi: 10.1016/j.techsoc.2019.04.001.

[31] C. M. Govender and M. Wait, "Work integrated learning benefits for student career prospects - mixed mode analysis," South African Journal of Universities, vol. 31, no. 5, Sep. 2017, doi: 10.20853/31-5-609.

[32] C. L. Obioha and I. van Zyl, "Gameful design for skills development for youths in urban marginalised communities," Interaction Design and Architecture(s), no. 53, pp. 27–50, Aug. 2022, doi: 10.55612/s-5002-053-002.

[33] J. A. van der Poll, "A Research Agenda for Embedding 4IR Technologies in the Leadership Management of Formal Methods," in 2022 International Conference on Computational Science and Computational Intelligence (CSCI), IEEE, Dec. 2022, pp. 1837–1842. doi: 10.1109/CSCI58124.2022.00331.

[34] J. M. Serumaga-Zake and J. A. van der Poll, "Addressing the impact of fourth industrial revolution on south african manufacturing small and medium enterprises (SMEs)," Sustainability (Switzerland), vol. 13, no. 21, Nov. 2021, doi: 10.3390/su132111703.

[35] E. Rose, J. Le Heron, and I. Sofat, "Student Understandings of Information Systems Design, Learning and Teaching: A Phenomenography: A Phenomenography Approach," Journal of Information Systems Education, vol. 16, no. 2, pp. 183–195, 2005.

[36] F. Marton, "Phenomenography: A research approach to investigating different understandings of reality," in Qualitative research in education. , Routledge, 1986, p. . 141-161.

[37] F. Han and R. A. Ellis, "Using Phenomenography to Tackle Key Challenges in Science Education," Front Psychol, vol. 10, Jun. 2019, doi: 10.3389/fpsyg.2019.01414.

[38] T. Stenfors‐Hayes, H. Hult, and M. A. Dahlgren, "A phenomenographic approach to research in medical education," Med Educ, vol. 47, no. 3, pp. 261‐270, Mar. 2013, doi: 10.1111/medu.12101.

[39] C. Yates, H. Partridge, and C. Bruce, "Exploring information experiences through phenomenography," Library and Information Research, vol. 36, no. 112, pp. 96–119, Sep. 2012, doi: 10.29173/lirg496.

[40] V. Braun and V. Clarke, "Using thematic analysis in psychology," Qual Res Psychol, vol. 3, no. 2, pp. 77–101, Jan. 2006, doi: 10.1191/1478088706qp063oa.

Development of a Curvature Sensing System using Fibre Bragg Grating Sensors.

Odendaal Mudondo
Department of Electrical and Electronic Engineering Science, University of Johannesburg, P.O.Box 624, Johannesburg,2000, South Africa
219063627@student.uj.ac.za

Michael F Grobler
Department of Electrical and Electronic Engineering Science, University of Johannesburg, P.O.Box 624, Johannesburg,2000, South Africa
michaelg@uj.ac.za

Natinale Yosef
Department of Electrical and Electronic Engineering Science, University of Johannesburg, P.O.Box 624, Johannesburg,2000, South Africa
202000323@student.uj.ac.za

***Abstract*—Structural Health Monitoring requires sensing systems that capture curvature and deformation in real time, particularly in environments prone to sudden instability. Fibre Bragg grating (FBG) sensors offer high sensitivity, but their wider adoption has been constrained by challenges in calibration, embedding within host materials, and effective data visualisation. This study reports the development of a real-time curvature sensor with embedded FBG arrays. The sensor was calibrated by bending it on known curvatures, utilising curvatures ranging from 0 to 0.02mm^{-1} in incremental steps to establish the relationship between wavelength shifts and curvature. A 60×30 cm experimental panel was fabricated for validation testing using progressive loading with weights of 100g, 200g, and 300g in a cantilever beam configuration. Data was acquired using a static industrial HBM FS22SI interrogator at a 1Hz sampling rate. Experimental evaluation demonstrated linear scaling behaviour with a maximum curvature value of 2.3×10^{-6} m^{-1} under a 300g loading, whilst the system captured curvatures as small as 1.5×10^{-7} mm^{-1}. In parallel, a React-based application was developed to enable real-time visualisation of the curvature fields, bridging raw optical data with immediate, interpretable outputs.**

Keywords—Curvature, Fibre Bragg Grating, React, Reconstruction

I. Introduction

The research and development of flexible shape sensors in multiple engineering domains, such as aerospace, soft robotics, and infrastructure monitoring, play a crucial role in real-time observation and collection of measured data related to the structural integrity of the monitored structure [1]. Fibre Bragg grating (FBG) sensors have attracted significant attention in recent years for curvature changes, owing to their high sensitivity and immunity to electromagnetic interference [2],[3]. Shape sensing using FBGs is increasingly recognised across diverse engineering disciplines, with key applications in structural health monitoring, early damage detection, and adaptive shape control, particularly in aerospace engineering [4]. Both contact and non-contact devices have been employed for curvature measurement; however, non-contact methods often suffer from delayed detection, as structural damage is only identified once deformation becomes significant. Conventional approaches, such as resistance strain gauges, are also prone to noise interference and have reduced reliability under dynamic conditions [5],[6]. Consequently, contact-based sensors such as FBGs offer distinct advantages for the early detection of curvature changes. This technology finds critical applications in diverse sectors [7]. Real-time shape monitoring capabilities provide essential feedback for structural design validation, operational testing, and performance optimisation, delivering diagnostic information that enables proactive maintenance and safety assessment. Conventional shape monitoring approaches typically depend on displacement or positional measurement techniques that require establishing fixed reference points in space, which can limit their applicability in dynamic or inaccessible environments [8]. The application of FBG sensors for curvature-based shape sensing has advanced over the past decade, driven by the demand for lightweight and high-resolution monitoring tools for engineering structures [9],[10]. The small size of optical fibre and fibre sensors, wavelength multiplexing capability, and immunity to electromagnetic interference make them highly suitable for applications in structural health monitoring (SHM), soft robotics, and aerospace systems [11],[12]. FBGs have been applied to measure beam deflections, monitor deformations in aircraft wings, and measure complex morphologies of soft and flexible components. Recent work has shown the value of FBGs for structural health monitoring (SHM). The FBG-based curvature sensor for bridge monitoring has demonstrated high accuracy in curvature reconstruction and reliable long-term operation in outdoor conditions [13],[14]. Similarly, reference [15] highlights a distributed FBG curvature sensor network for composite aircraft structures, emphasising its adaptability for large-scale infrastructure applications compared to traditional point-based sensors [16]. FBG sensors were used to monitor offshore wind turbine foundations. They achieved stability in curvature measurement over 18 months of continuous monitoring in a marine environment. An FBG-based monitoring system for the wind turbine blade root section was implemented, achieving real-time curvature tracking with sub-millimetre deflection accuracy and demonstrating its use in critical rotating machinery [17]. However, these applications focused on rigid structural materials such as steel and concrete, with limited exploration of flexible composite substrates [18]. A significant trend in recent literature has been the adoption of multi-core fibres (MCFs) for enhanced three-dimensional shape reconstruction capabilities. MCFs were utilised for precise curvature measurement in medical catheter applications, achieving improved accuracy by measuring strain in multiple orientations within a single fibre structure. The authors reported improvements of up to 40% in position accuracy compared to single-core implementations [19]. Building upon this foundation, the MCF approach was extended to continuum robotic applications, where multi-core FBGs enabled precise 3D reconstruction under complex bending and twisting conditions with angular accuracy [20]. Further advancement in MCF technology was demonstrated by exploring four-core fibres for enhanced spatial resolution in soft robotics applications, while [21] investigated twisted multi-core configurations for improved bending sensitivity. Advanced signal processing techniques for MCF data were developed by [22], who implemented Kalman filtering algorithms for real-time shape reconstruction from multi-core FBG arrays. Researchers have extensively experimented with embedding FBGs into flexible substrates. In [23], FBG-

embedded silicone sensors for soft robotics were developed, achieving a curvature sensitivity of 1×10^{-3} m^{-1} with minimal hysteresis. The development of integrated digital platforms for FBG monitoring systems remains significantly limited [24]. Most research focuses on sensor development with minimal attention to user-friendly, real-time monitoring interfaces. Limited work includes basic web-based platforms restricted to laboratory conditions and desktop applications lacking real-time capabilities [14],[15]. Integrating modern web technologies with FBG interrogation systems represents a significantly underexplored area [16].

The present research lies in addressing these gaps by:

- A single-core FBG curvature sensor embedded in a flexible fibreglass substrate.
- Establishing a curvature–temperature calibration framework to decouple strain and thermal effects.
- Implementing a React-based real-time monitoring platform interfaced with the FS22si interrogator for live visualisation, data streaming, and analysis.

Combining a single-core approach and real-time digital integration, this work presents a practical and deployable FBG-based shape sensing solution that extends beyond the laboratory to field applications in structural health monitoring.

II. Operating Principle of Fibre Bragg Gratings

A Fibre Bragg Grating is a periodic variation of the refractive index along the core of an optical fibre. The gratings are fabricated using the phase mask method and a 266 nm wavelength Nd: YAG laser. This periodic pattern acts as a partial mirror; when broadband light propagates through the grating, only the wavelength that satisfies the Bragg condition is reflected, while the other wavelengths continue to propagate through the grating structure. The Bragg condition is defined as:

$$\lambda_B = 2n_{eff}\Delta \tag{1}$$

λ_B is the Bragg wavelength, n_{eff} is the effective refractive index of the fibre core, and Δ is the spacing of the grating period [7]. Variations in mechanical strain or temperature change either the period of the grating or the refractive index, resulting in a measurable shift of the reflected wavelength. The general expression for this wavelength shift is:

$$\Delta\lambda_B = \lambda_B[(1-P_e)\varepsilon + (\alpha+\xi)\Delta T] \tag{2}$$

P_e is the effective photoelastic coefficient, ε is the strain, α is the thermal expansion coefficient, and ξ represents the thermo-optic coefficient. This equation shows that strain and temperature influence the Bragg wavelength [25].

The equation below mathematically describes the relation between the Bragg wavelength, strain and temperature.

$$\frac{\Delta\lambda_B}{\lambda_B} = K_\varepsilon\varepsilon + \xi\Delta T \tag{3}$$

The Bragg wavelength shifts due to applied strain (ε) and temperature variations (T)[25]. In terms of strain λ_B=λ_s

$$\frac{\Delta\lambda_s}{\lambda_s} = K_\varepsilon\varepsilon \tag{4}$$

Thus, the above equation is when temperature is excluded or is kept constant [7]. If the strain is excluded, then:

$$\frac{\Delta\lambda_t}{\lambda_t} = \xi\Delta T \tag{5}$$

Temperature compensation strategies are commonly employed to separate mechanical strain from thermal effects, such as using reference gratings isolated from strain [7].

When an FBG is surface-mounted or embedded in a host structure, bending causes the fibre to experience tensile strain on one side and compressive strain on the other if it is offset from the central axis of the material in which it is embedded. Since strain varies linearly with the distance from the neutral axis of the material in which it is embedded, curvature κ can be derived from the measured wavelength shift. Assuming temperature compensation has been applied, the relation between curvature and the Bragg wavelength shift is given by:

$$k(t) = \frac{1}{z_s(1-P_e)}\frac{\Delta\lambda_B(t)}{\lambda_B} \tag{6}$$

Where z_s is the vertical distance of the fibre from the neutral axis, λ_B is the reference Bragg wavelength, and $\Delta\lambda_B(t)$is the time-varying wavelength shift [8].

III. Methodology

A. Sensor Design

The sensor was made from a boron germanium co-doped photosensitive fibre (FibreCore PS1250/1500) consisting of two single-core optical fibres, each with five FBGs, embedded in a 60×30cm fibreglass strip, as illustrated in Figure 1. The fibres are placed parallel, 12.5 cm apart, with five gratings along each fibre. They are positioned at a depth of 1.25 mm below the surface, covered by an additional 0.25 mm fibreglass layer to ensure protection and strain transfer. This arrangement provides detectable curvature measurements and reconstructs bending information across the strip [26].

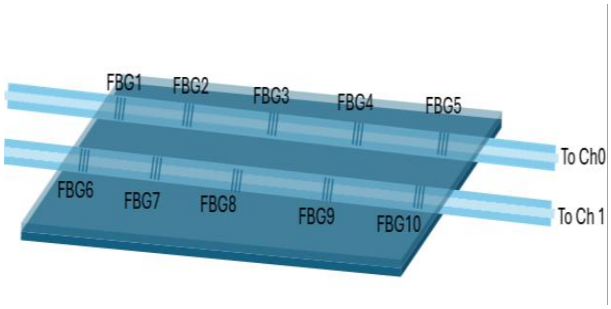


Figure 1: Sensor Design

Figure 1 shows the arrangement of five FBGs on Fibre 1 (Ch 0) and five FBGs on Fibre 2 (Ch 1), placed at distances of 3, 18, 31, 45, and 58 cm from one side along the fibreglass substrate. The reflected wavelengths of the FBGs are shown in Table 1, which were obtained after embedding them into the fibreglass and epoxy. This confirms that all gratings remained functional following integration and serve as reference points for curvature measurements.

The paired gratings at each position provide corresponding Bragg wavelengths, enabling curvature measurements for shape reconstruction.

TABLE 1: PLACEMENT OF FBGS ALONG FIBRE 1 (CH 0) AND FIBRE 2 (CH 1)

FBG Position	Distance from Reference End (cm)	Fibre 1(Ch 0) Wavelength	Fibre 1(Ch 1) Wavelength
FBG 1	3	1531.576	1529.450
FBG 2	18	1555.877	1556.287
FBG 5	31	1536.190	1535.913
FBG 3	45	1561.729	1561.494
FBG 4	58	1544.692	1544.704

B. Temperature Calibration

Temperature calibration was conducted to establish the relationship between wavelength shift and temperature variations while eliminating strain-induced wavelength shift. The following equipment was used:

- Yokogawa AG637D 600-1700nm Optical Spectrum Analyser
- Superlum Superluminescent diode
- Digital thermometer
- Series 2000 Scientific oven
- FBG at a 1544.3nm wavelength

The experimental setup is shown in Figure 2. An FBG was attached in a fixed configuration to the Series 2000 Scientific oven, isolating it from external strain. The oven's temperature varied across two distinct ranges: initially from 35 to 70 °C and then from 100 to 200 °C. The shift in wavelength was recorded for every degree change made. The fundamental principle underlying FBG temperature sensitivity is governed by two primary mechanisms: thermal expansion of the fibre and the thermo-optic effect.

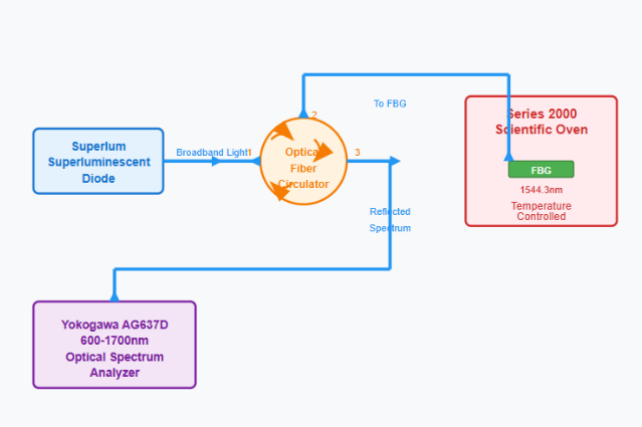


Figure 2: Temperature calibration setup

The results obtained from the experiment show that as the temperature increases, the shift in wavelength also increases. This illustrates a linear relationship between temperature and wavelength change, while the strain effect is isolated.

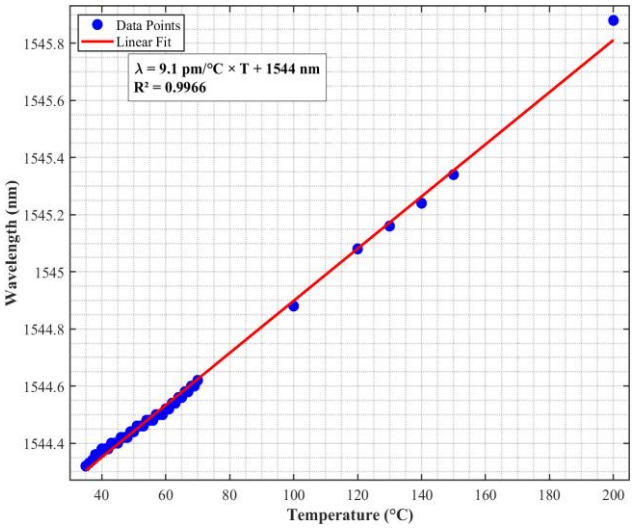


Figure 3: Temperature Calibration: Wavelength vs Temperature

The experimental results demonstrated a linear relationship between temperature and wavelength shift, with a wavelength shift of 9.1 pm/°C, which falls within the range of values found in literature [27], [28].

C. Curvature Calibration

The curvature was calibrated by maintaining a constant ambient laboratory temperature to eliminate thermal effects on the FBG response. This ensured that the temperature effect was isolated in the reaction of the FBGs, and the mechanical strain was solely causing the wavelength shift.

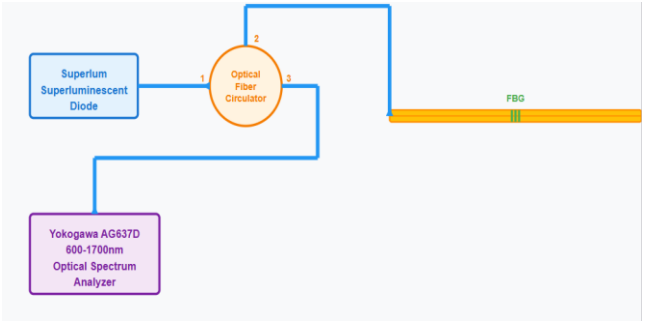


Figure 4: Curvature calibration setup

The FBG was embedded at an offset from the centre of the fibreglass composite substrate to ensure unidirectional strain transfer and to distinguish bending direction. This ensured that the FBG experienced measurable tension or compression during curvature application, which indicates the bending direction. It also improves sensitivity and eliminates other variables, such as torsion-induced effects, as they could introduce measurement errors during calibration [7].

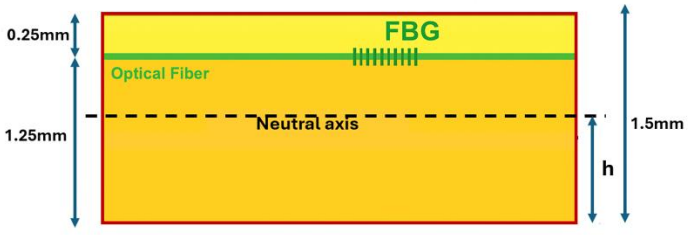


Figure 5: Cross-section of fibreglass composite strip with embedded FBG.

This configuration restricts the strain-induced curvature to only one plane, providing a controlled environment for accurate calibration. The following equipment was used:

- Yokogawa AG637D 600-1700nm Optical Spectrum Analyser
- Superlum Super luminescent Diode
- FBG embedded in fibreglass

The calibration procedure utilised segments of known radii ranging from 0 to 350 mm. The curvature-strain relationship follows fundamental beam theory:

$$\varepsilon = \frac{h.k}{2} \quad (4)$$

Where $\boldsymbol{\varepsilon}$ is the induced strain, h is the distance from the neutral axis, and k is the curvature applied to the FBG. Figure 6 shows the experimental setup used, and Figure 8 shows the curvature calibration curves.

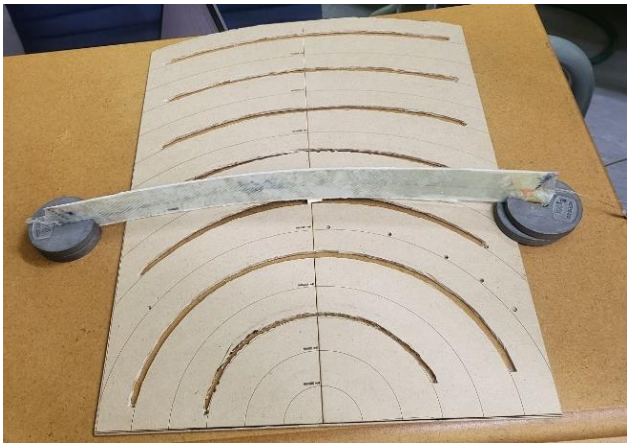


Figure 6: Calibration block

Figure 7 show the bending direction of the FBGs. When the fibreglass strip is bent in a convex orientation (curving downward from the 0.25mm layer side), it results in a negative wavelength shift, as shown in Figure 8, the grating period decreases, and this direction of curvature is negative, and the grating experiences compression [29]. When the strip is bent in a concave orientation (curving upward to the 1.25mm layer side), a positive wavelength shift occurs due to the grating's period expansion. This curvature direction is positive, and the FBG is under tension. [29]

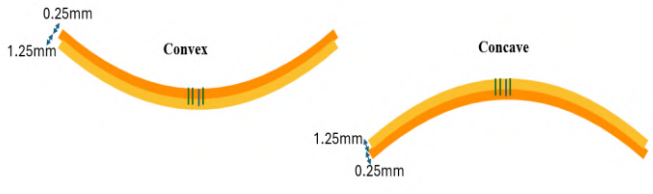


Figure 7: Bending orientations

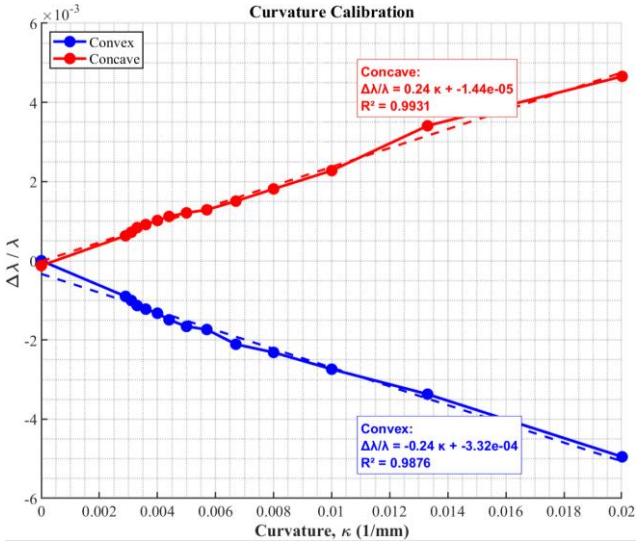


Figure 8: Curvature Calibration: Wavelength vs Curvature.

The experimental results in Figure 8 demonstrate a linear curvature response between the applied curvature and wavelength shift. Calibration data, presented in the red and blue graphs, reveals the sensor's dual-response behaviour, characterised by both positive and negative wavelength shifts [30]. The red graph represents the sensor under tension, exhibiting a positive wavelength shift with a curvature constant of +0.24mm. The blue graph illustrates the response under compression, demonstrating a negative change with a gradient of -0.24mm. The distinction between these responses confirms the sensor's ability to differentiate between different bending directions, whether convex (compression) or concave(tension) [31].

The calibration methodology was adapted from established curvature sensing principles documented in literature, particularly the work on FBG-based curvature sensors embedded in 3D printed patches and dual FBG shape sensor configurations [4], [32], [33]. The calibration results confirm that the sensor exhibits predictable, repeatable behaviour across the operational curvature range, with minimal hysteresis effects. The correlation between theoretical predictions and experimental measurements validates both the sensor design approach and the calibration [29].

IV. RESULTS AND DISCUSSION

The calibration results demonstrated a reliable response to temperature and curvature measurements. The sensor exhibited a temperature sensitivity of 9.1 pm/°C, which falls within the range reported in the literature [27], [28]. A good linear relationship was observed between temperature and wavelength shift, with a correlation coefficient (R^2) of 0.9966. The curvature calibration, as illustrated in Figure 6, with a known curvature ranging from 0 to 0.02 mm^{-1}, the sensor detects curvature variations under both tension and compression. This performance was characterised by a curvature constant of approximately ±4.17 mm^{-1} and a linear correlation coefficient of 0.9931 under tension and 0.9876 under compression loading, as shown in Figure 8, thereby validating the capability of the developed FBG sensor for accurate curvature monitoring [30].

The experimental setup consisted of a 60×30 cm fibreglass plate with FBGs embedded. One edge of the plate was

clamped to impose a cantilever boundary condition. This setup facilitated upward and downward deflections of the fibreglass plate, which were subsequently characterised as curvature through the calibrated response of the FBG sensors. The experimental procedure involved 30 seconds without weight, 120 seconds with weight applied, and 30 seconds after the weight was removed. This approach captures the rapid changes that occur when loads are applied or removed, as well as the stable readings during loading.

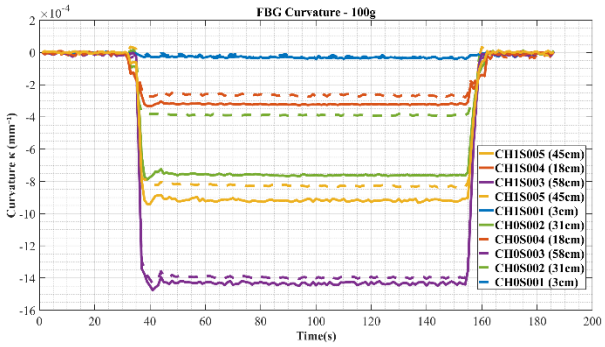


Figure 9: FBGs under compression with a 100g weight

The FBGs at 58cm (CH1S003 and CH0S003, shown in purple) exhibit the maximum compression response, reaching steady-state curvature values of -14×10^{-7} mm^{-1}. This position, closest to the free end where maximum deflection occurs, experiences the largest curvature. The individual FBG curvature values when loaded with a 100g are shown in Table 2.

TABLE 2: FBG CURVATURE RESPONSE UNDER 100G COMPRESSION LOADING

Position	CH0 FBG	CH0 Curvature ($\times10^{-7}$ mm^{-1})	CH1 FBG	CH1 Curvature ($\times10^{-7}$ mm^{-1})
58cm	CH1S003	-14.0	CH0S003	-14.0
45cm	CH1S005	-8.0	CH0S005	-9.0
31cm	CH1S002	-4.0	CH0S002	-7.5
18cm	CH1S004	-3.0	CH0S004	-3.0
3cm	CH1S001	0	CH0S001	0

The FBGs demonstrate rapid response within 2-3 seconds of load application, achieving steady-state values with minimal overshoot and negligible drift during the 120-second loading period. Upon load removal at t = 150s, all FBGs return to their reference state, confirming elastic deformation and validating the shape reconstruction capability. The method shows that the structure returns to its original shape and that the FBGs can monitor the loading and unloading processes for real-time structural monitoring.

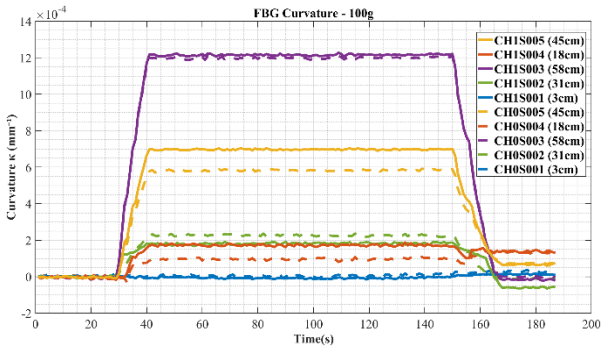


Figure 10: FBGs under tension with a 100g weight

Figure 10, demonstrates the tension response of the FBGs under 100g loading, exhibiting positive curvature values throughout the measurement period. The FBGs follow the same position-dependent gradient observed in compression testing, with maximum curvature occurring at the 58cm position (CH1S003/CH0S003) reaching 12.0×10^{-7} m^{-1}. The transient response characteristics demonstrate the compression behaviour, with rapid stabilisation within 2-3 seconds of load application at t = 30s and a complete return to zero upon load removal at t = 150s.

TABLE 3: FBG CURVATURE RESPONSE UNDER 100G TENSIONAL LOADING

Position	CH0 FBG	CH0 FBG Curvature ($\times10^{-7}$ mm^{-1})	CH1 FBG	CH1 Curvature ($\times10^{-7}$ mm^{-1})
58cm	CH1S003	12.0	CH0S003	12.0
45cm	CH1S005	7.0	CH0S005	5.8
31cm	CH1S002	2.2	CH0S002	2.0
18cm	CH1S004	1.8	CH0S004	1.5
3cm	CH1S001	0	CH0S001	0

The curvature magnitudes progressively decrease from the free end towards the fixed support, with FBGs at a 3cm position showing negligible response. The individual FBG curvature values when loading a 100g under tension are shown in Table 3 CH1 and CH0 FBGs demonstrate good agreement across most positions, showing structural symmetry in the tension loading configuration

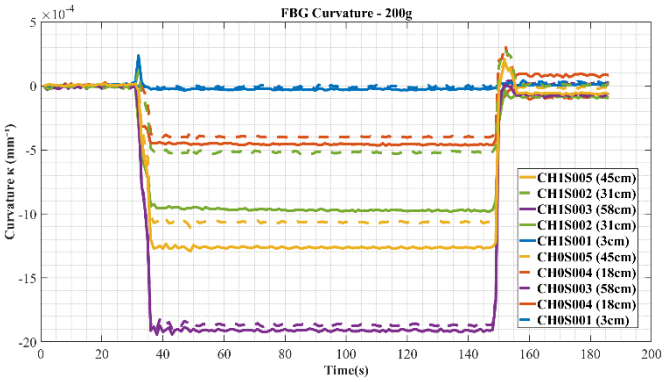


Figure 11: FBGs under compression with a 200g weight

TABLE 4: FBG CURVATURE RESPONSE UNDER 200G COMPRESSION LOADING

Position	CH0 FBG	CH0 FBG Curvature ($\times10^{-7}$ mm^{-1})	CH1 FBG	CH1 Curvature ($\times10^{-7}$ mm^{-1})
58cm	CH1S003	-19.0	CH0S003	-19.0
45cm	CH1S005	-13.0	CH0S005	-12.5
31cm	CH1S002	-10.0	CH0S002	-9.5
18cm	CH1S004	-4.5	CH0S004	-4.2
3cm	CH1S001	0	CH0S001	0

The 200g compression loading exhibits a maximum curvature of -19.0×10^{-4} m^{-1} at 58 cm, which is double the value obtained with 100g. The individual FBG curvature values when loading 200g under compression are shown in Table 4. The transient response remains consistent, with rapid stabilisation occurring within 2-3 seconds. The linear sensor

response and good correlation between CH1 and CH0 validate measurement consistency under increased loading conditions.

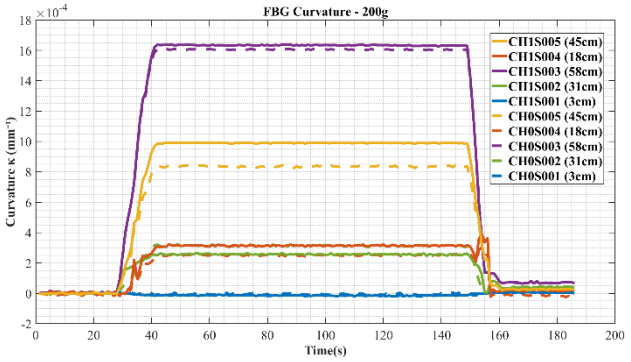


Figure 12: FBGs under tension with a 200g weight

The 200g tension loading, as shown in Figure 12, exhibits with maximum positive curvature at 58cm reaching 16.0×10^{-7} mm^{-1}, maintaining the expected linear response characteristics.

The position-dependent gradient follows the established pattern with progressive decrease towards the fixed support. Transient behaviour demonstrates rapid stabilisation within 2-3 seconds and a complete return to the reference level upon load removal, confirming elastic deformation and excellent shape recovery capability under doubled loading conditions. The individual FBG curvature values when loading a 200g under tension are shown in Table 5.

TABLE 5: FBG CURVATURE RESPONSE UNDER 200G TENSIONAL LOADING

Position	CH0 FBG	CH0 Curvature ($\times10^{-7}$ mm^{-1})	CH1 FBG	CH1 Curvature ($\times10^{-7}$ mm^{-1})
58cm	CH1S003	16.0	CH0S003	16.0
45cm	CH1S005	10.0	CH0S005	9.8
31cm	CH1S002	4.0	CH0S002	3.8
18cm	CH1S004	3.2	CH0S004	3.0
3cm	CH1S001	0	CH0S001	0

Figure 13, illustrates 300g compression loading demonstrates continued linear scaling with maximum curvature at 58cm reaching -2.3×10^{-7} mm^{-1}, representing a threefold increase from the 100g baseline. The tabulated results show the characteristic position-dependent gradient, with curvature values decreasing systematically from -2.3×10^{-7} mm^{-1}at 58cm to -1.5×10^{-7} mm^{-1} at 45cm, -1.0×10^{-7} mm^{-1}at 31cm, and -0.6×10^{-7} mm^{-1} at 18cm, whilst FBGs at 3cm remain near zero.

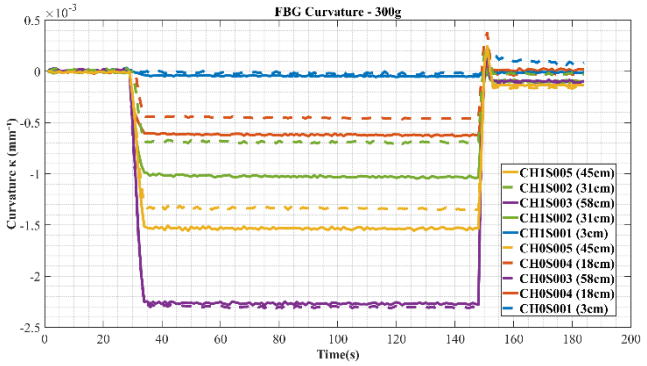


Figure 13: FBGs under compression with a 300g weight

TABLE 6: FBG CURVATURE RESPONSE UNDER 300G COMPRESSION LOADING

Position	CH0 FBG	CH0 Curvature ($\times10^{-7}$ m^{-1})	CH1 FBG	CH1 Curvature ($\times10^{-7}$ m^{-1}
58cm	CH1S003	-2.3	CH0S003	-2.3
45cm	CH1S005	-1.5	CH0S005	-1.4
31cm	CH1S002	-1.0	CH0S002	-1.0
18cm	CH1S004	-0.6	CH0S004	-0.5
3cm	CH1S001	0	CH0S001	0

.

Table 6 , shows the individual FBG curvature values when loading a 300g specimen under compression. The sensor array maintains rapid transient stabilisation and returns reference level upon unloading, with excellent agreement between CH1 and CH0 sensor pairs across all positions, confirming measurement reliability under maximum loading conditions.

The 300g tension loading shown in Figure 14 , demonstrates maximum positive curvature at 58cm, reaching 2.2×10^{-7} mm^{-1}, maintaining the threefold scaling relationship from the 100g baseline. The tabulated results in Table 7, reveal the systematic position-dependent distribution with values decreasing from 2.2×10^{-7} mm^{-1}at 58cm to 1.4×10^{-7} mm^{-1}at 45cm, 0.6×10^{-7} mm^{-1}at 31cm, and 0.4×10^{-7} mm^{-1}at 18cm, whilst FBGs at 3cm remain negligible.

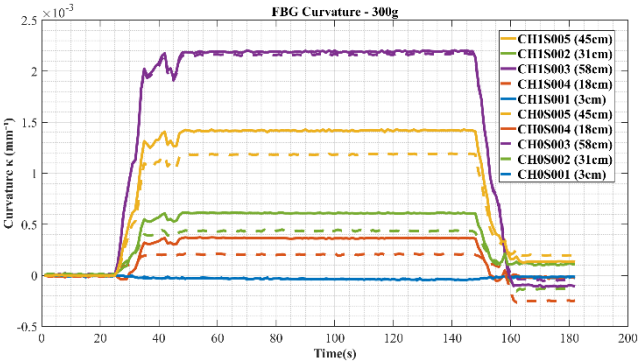


Figure 14: FBGs under tension with a 300g weight

TABLE 7: FBG CURVATURE RESPONSE UNDER 300G TENSIONAL LOADING

Position	CH0 FBG	CH0 Curvature ($\times10^{-6}$ mm^{-1})	CH1 FBG	CH1 Curvature ($\times10^{-6}$ mm^{-1}
58cm	CH1S003	2.2	CH0S003	2.2
45cm	CH1S005	1.4	CH0S005	1.2
31cm	CH1S002	0.6	CH0S002	0.6
18cm	CH1S004	0.4	CH0S004	0.3
3cm	CH1S001	0	CH0S001	0

The progressive loading tests, using weights of 100g, 200g, and 300g, demonstrate the linear response characteristics and shape reconstruction capabilities of the FBG sensor array. Data acquisition was performed using a static industrial HBM FS22SI interrogator operating at a sampling rate of 1Hz, providing adequate resolution for capturing the curvature response dynamics. The curvature measurements exhibit linear scaling across the loading range, with values at the 58cm position scaling proportionally from -14.0×10^{-7} mm^{-1} (100g compression) to -19.0×10^{-7} mm^{-1} (200g) and -2.3×10^{-7} mm^{-1} (300g). Under tension, similar linearity is demonstrated

with values of 12.0×10^{-7} mm^{-1},16.0×10^{-7} mm^{-1}, and 2.2×10^{-7} mm^{-1}, respectively.
The curvature distribution consistently follows cantilever beam theory across all loading conditions, with maximum values occurring at the free end (58cm) and progressively decreasing towards the fixed support. The 300g loading condition illustrates this gradient with curvature values of -2.3×10^{-6} mm^{-1} at 58cm, -1.5×10^{-6} mm^{-1} at 45cm, -1.0×10^{-6} mm^{-1} at 31cm, -0.6×10^{-6} mm^{-1} at 18cm, and negligible response at 3cm. This curvature distribution enables accurate reconstruction of the complete deflection profile along the cantilever length, providing the discrete curvature data points necessary for shape sensing algorithms [34].

The progressive loading in Figure 15, graphs show stepwise curvature increases with each 100g weight addition at 60-second intervals.

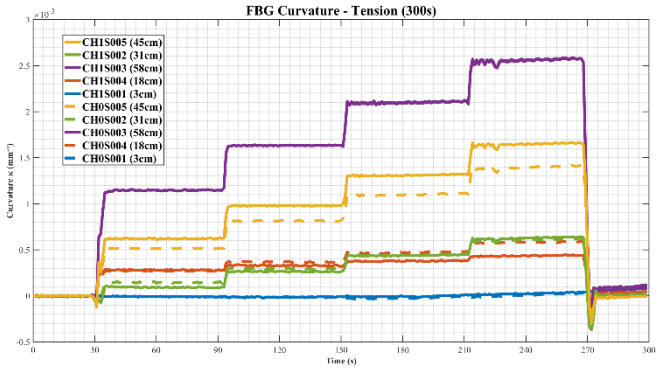


Figure 15: FBGs under tension with 100g,200g,300g and 400g weights

The 58cm sensor (CH1S002) exhibits distinct curvature steps: approximately 3×10^{-6} mm^{-1} after the first 100g, 6×10^{-6} mm^{-1} after 200g, 9×10^{-6} mm^{-1} after 300g, and 12×10^{-6} mm^{-1} after a total loading of 400 g. The 45cm sensor (CH0S005) follows a similar pattern with proportionally smaller magnitudes, reaching approximately 8×10^{-6} mm^{-1} at maximum loading. Each loading step produces an immediate curvature response with stable plateaus during the 60-second hold periods, whilst sensors at 31cm and 18cm positions exhibit correspondingly smaller incremental increases. Upon complete load removal at t = 240s, all sensors return to zero within seconds, confirming elastic behaviour and demonstrating the system's capability for monitoring cumulative structural loading and real-time deformation tracking in progressive loading scenarios.

The FBG sensors demonstrate rapid response characteristics with stabilisation occurring within 2-3 seconds of load application across all weight conditions. This fast response capability highlights the inherent advantage of optical fibre sensing for real-time structural monitoring applications, with the rapid transient behaviour combined with immediate return to baseline upon load removal confirming the system's capability for dynamic shape reconstruction. The cantilever beam configuration provides an ideal validation platform for demonstrating FBG shape sensing capabilities, with defined boundary conditions and predictable mechanical response. The experimental procedure, consisting of 30 seconds without weight, 120 seconds of loading, and 30 seconds of recovery, captures both transient and steady-state responses while demonstrating complete shape recovery capabilities. Future work shall integrate machine learning algorithms to enhance shape reconstruction accuracy and develop predictive capabilities for structural behaviour under varying loading conditions



Figure 16: Compression React- based shape reconstruction

Figure 16, and Figure 17, illustrate 2D shape reconstruction on a React-based dashboard, providing a complementary perspective to the curvature-time graphs. Figure 16, shows a shape reconstruction when a 300g compressional weight was

applied at the free end, and Figure 17 shows the shape reconstruction when a 300g tensional load was applied at the free end. These two-dimensional profiles of the sensor, reconstructed in real time, visually confirm the numerical curvature results.



Figure 17: Tension React- based shape reconstruction

The real-time shape sensing system was implemented using a React-based monitoring interface with the HBM FS22SI Industrial BraggMETER SI through TCP/IP communication protocols. The system leverages Python libraries, including for real-time data visualisation and NumPy for signal processing operations, operating at a 1 Hz sampling rate as specified by the HBM interrogator's configuration. The architecture establishes a TCP connection to stream wavelength shift data from individual FBG sensors positioned across the flexible substrate, enabling continuous monitoring of structural deformation states [35]. The Python implementation processes incoming data streams to compute curvature values, where mechanical loading induces measurable wavelength shifts that are subsequently converted to curvature. The interface plots demonstrate the system's ability to distinguish between directional loading, as illustrated in Figure 16, and in Figure 17, through real-time monitoring. This system configuration facilitates real-time visualisation of bending scenarios, with the matplotlib-based interface providing immediate feedback on structural deformation patterns across the fibreglass FBG array. Dual numerical and visual validation represents a good step in linking optical fibre sensing to real-world monitoring and analysis [35].

Conclusion

This study demonstrated the development of a real-time fibre Bragg grating (FBG) curvature sensor for structural monitoring and data-driven analysis. By embedding dual FBG arrays in a 60 × 30 cm composite strip and applying baseline-corrected calibration for strain and temperature, the system captured wavelength shifts and transformed them into curvature values. A maximum curvature value of 2.3×10^{-6} mm^{-1} under a 300g loading was obtained, whilst the system captured curvatures as small as 1.5×10^{-7} mm^{-1}. The React interface displays a 2D reconstruction, shown in green, of the cantilever beam, highlighting the variations in curvature.

This work integrates the sensing framework with a React-based application for real-time 2D shape reconstruction. The findings highlight the potential of this FBG curvature sensor for structural health monitoring, where accurate detection of local bending and real-time visualisation are critical. The large amount of data collected could be utilised to enhance structural health monitoring in complex structures.

Acknowledgment

This work was supported by the CSIR African Laser Centre, Telkom, the National Laser Centre (NLC), and the Photonics Research Centre at the University of Johannesburg.

References

[1] A. Mardanshahi, A. Sreekumar, X. Yang, S. K. Barman, and D. Chronopoulos, “Sensing Techniques for Structural Health Monitoring: A State-of-the-Art Review on Performance Criteria and New-Generation Technologies,” *Sensors 2025, Vol. 25, Page 1424*, vol. 25, no. 5, p. 1424, Feb. 2025, doi: 10.3390/S25051424.

[2] F. Zhu *et al.*, “Stress-insensitive vector curvature sensor based on a single fiber Bragg grating,” *Optical Fiber Technology*, vol. 54, p. 102133, Jan. 2020, doi: 10.1016/J.YOFTE.2019.102133.

[3] K. P. Wong, H. T. Kim, A. Wang, K. Kiger, J. H. Duncan, and M. Yu, "High-speed interrogation of multiplexed fiber Bragg grating sensors for monitoring dynamic strain responses during a flexible plate impact on a water surface," *Smart Mater Struct*, vol. 29, no. 12, p. 125021, Nov. 2020, doi: 10.1088/1361-665X/ABC23A.

[4] H. Wu, L. Liang, H. Wang, S. Dai, Q. Xu, and R. Dong, "Design and Measurement of a Dual FBG High-Precision Shape Sensor for Wing Shape Reconstruction," *Sensors 2022, Vol. 22, Page 168*, vol. 22, no. 1, p. 168, Dec. 2021, doi: 10.3390/S22010168.

[5] K. Huang, J. Yan, L. Zhang, F. Zhang, M. Jiang, and Q. Sui, "Reconstruction Technology of Flexible Structure Shape Based on FBG Sensor Array and Deep Learning Algorithm," *SDHM Structural Durability and Health Monitoring*, vol. 16, no. 2, pp. 179–194, Apr. 2022, doi: 10.32604/SDHM.2022.018202.

[6] X. Zhang, Y. Song, F. Meng, G. Sun, and L. Zhu, "Optical fiber based soft curvature sensor with polyvinyl chloride reinforced silicone rubber substrate," *Optik (Stuttg)*, vol. 178, pp. 567–574, Feb. 2019, doi: 10.1016/J.IJLEO.2018.09.128.

[7] A. N. D. Alhussein, M. R. T. M. Qaid, T. Agliullin, B. Valeev, O. Morozov, and A. Sakhabutdinov, "Fiber Bragg Grating Sensors: Design, Applications, and Comparison with Other Sensing Technologies," *Sensors 2025, Vol. 25, Page 2289*, vol. 25, no. 7, p. 2289, Apr. 2025, doi: 10.3390/S25072289.

[8] jingying wu *et al.*, "Fiber Bragg grating-based shape sensing: a review and perspective," p. 84, Mar. 2021, doi: 10.1117/12.2590684.

[9] J. K. Sahota, N. Gupta, and D. Dhawan, "Fiber Bragg grating sensors for monitoring of physical parameters: a comprehensive review," *https://doi.org/10.1117/1.OE.59.6.060901*, vol. 59, no. 6, p. 060901, Jun. 2020, doi: 10.1117/1.OE.59.6.060901.

[10] M. F. Bado and J. R. Casas, "A Review of Recent Distributed Optical Fiber Sensors Applications for Civil Engineering Structural Health Monitoring," *Sensors 2021, Vol. 21, Page 1818*, vol. 21, no. 5, p. 1818, Mar. 2021, doi: 10.3390/S21051818.

[11] T. L. T. Lun, K. Wang, J. D. L. Ho, K. H. Lee, K. Y. Sze, and K. W. Kwok, "Real-Time Surface Shape Sensing for Soft and Flexible Structures Using Fiber Bragg Gratings," *IEEE Robot Autom Lett*, vol. 4, no. 2, pp. 1454–1461, Apr. 2019, doi: 10.1109/LRA.2019.2893036.

[12] E. A. Souza, L. C. Macedo, A. Frizera, C. Marques, and A. Leal-Junior, "Fiber Bragg Grating Array for Shape Reconstruction in Structural Elements," *Sensors 2022, Vol. 22, Page 6545*, vol. 22, no. 17, p. 6545, Aug. 2022, doi: 10.3390/S22176545.

[13] C. Xu and Z. S. Khodaei, "A Novel Fabry-Pérot Optical Sensor for Guided Wave Signal Acquisition," *Sensors 2020, Vol. 20, Page 1728*, vol. 20, no. 6, p. 1728, Mar. 2020, doi: 10.3390/S20061728.

[14] C. Y. Hong, Y. F. Zhang, M. X. Zhang, L. M. G. Leung, and L. Q. Liu, "Application of FBG sensors for geotechnical health monitoring, a review of sensor design, implementation methods and packaging techniques," *Sens Actuators A Phys*, vol. 244, pp. 184–197, Jun. 2016, doi: 10.1016/J.SNA.2016.04.033.

[15] Y. Zhang *et al.*, "Advances in sensors technologies for composite structural health monitoring," *Compos Struct*, vol. 370, p. 119448, Oct. 2025, doi: 10.1016/J.COMPSTRUCT.2025.119448.

[16] X. Yaoda *et al.*, "Integrated sensing technology for lithium ion battery," *Energy Storage Science and Technology*, vol. 11, no. 6, p. 1834, Jun. 2022, doi: 10.19799/J.CNKI.2095-4239.2022.0183.

[17] M. Liang, X. Fang, S. Li, G. Wu, M. Ma, and Y. Zhang, "A fiber Bragg grating tilt sensor for posture monitoring of hydraulic supports in coal mine working face," *Measurement*, vol. 138, pp. 305–313, May 2019, doi: 10.1016/J.MEASUREMENT.2019.02.060.

[18] D. Kinet, P. Mégret, K. W. Goossen, L. Qiu, D. Heider, and C. Caucheteur, "Fiber Bragg Grating Sensors toward Structural Health Monitoring in Composite Materials: Challenges and Solutions," *Sensors 2014, Vol. 14, Pages 7394-7419*, vol. 14, no. 4, pp. 7394–7419, Apr. 2014, doi: 10.3390/S140407394.

[19] P. S. Westbrook *et al.*, "Continuous Multicore Optical Fiber Grating Arrays for Distributed Sensing Applications," *Journal of Lightwave Technology*, vol. 35, no. 6, pp. 1248–1252, Mar. 2017, doi: 10.1109/JLT.2017.2661680.

[20] X. An *et al.*, "Shape reconstruction of soft continuum robots via the fusion of local strains and global poses," *Cell Rep Phys Sci*, vol. 5, no. 10, p. 102224, oct. 2024, doi: 10.1016/J.XCRP.2024.102224.

[21] A. Beisenova, A. Issatayeva, I. Iordachita, W. Blanc, C. Molardi, and D. Tosi, "Distributed fiber optics 3D shape sensing by means of high scattering NP-doped fibers simultaneous spatial multiplexing," *Opt Express*, vol. 27, no. 16, p. 22074, Aug. 2019, doi: 10.1364/OE.27.022074,.

[22] F. Khan, A. Denasi, D. Barrera, J. Madrigal, S. Sales, and S. Misra, "Multi-Core Optical Fibers with Bragg Gratings as Shape Sensor for Flexible Medical Instruments," *IEEE Sens J*, vol. 19, no. 14, pp. 5878–5884, Jul. 2019, doi: 10.1109/JSEN.2019.2905010.

[23] A. Leal-Junior *et al.*, "Fiber Bragg Gratings in CYTOP Fibers Embedded in a 3D-Printed Flexible Support for Assessment of Human–Robot Interaction Forces," *Materials 2018, Vol. 11, Page 2305*, vol. 11, no. 11, p. 2305, Nov. 2018, doi: 10.3390/MA11112305.

[24] C. E. Campanella, A. Cuccovillo, C. Campanella, A. Yurt, and V. M. N. Passaro, "Fibre Bragg Grating Based Strain Sensors: Review of Technology and Applications," *Sensors 2018, Vol. 18, Page 3115*, vol. 18, no. 9, p. 3115, Sep. 2018, doi: 10.3390/S18093115.

[25] M. Majumder, T. K. Gangopadhyay, A. K. Chakraborty, K. Dasgupta, and D. K. Bhattacharya,

"Fibre Bragg gratings in structural health monitoring-Present status and applications," *Sens Actuators A Phys*, vol. 147, no. 1, pp. 150–164, Sep. 2008, doi: 10.1016/j.sna.2008.04.008.

[26] W. S. Yang, C. C. Chiang, and H. C. Hsu, "Monitoring torque in bolts using an embedded fiber Bragg grating sensor," *Optik (Stuttg)*, vol. 291, p. 171294, Nov. 2023, doi: 10.1016/J.IJLEO.2023.171294.

[27] Q. Xia, X. Zhang, H. Wang, L. Yuan, and T. Yuan, "Wearable Glove with Enhanced Sensitivity Based on Push–Pull Optical Fiber Sensor," *Biosensors 2025, Vol. 15, Page 414*, vol. 15, no. 7, p. 414, Jun. 2025, doi: 10.3390/BIOS15070414.

[28] A. Aliberti *et al.*, "Microgel assisted Lab-on-Fiber Optrode," *Sci Rep*, vol. 7, no. 1, Dec. 2017, doi: 10.1038/S41598-017-14852-5.

[29] P. Di Palma, A. Iadicicco, and S. Campopiano, "Curvature Sensor Based on FBGs Embedded in 3D Printed Patches," *IEEE Sens J*, vol. 21, no. 16, pp. 17868–17874, Aug. 2021, doi: 10.1109/JSEN.2021.3083961.

[30] X. Yi, X. Cheng, F. Shi, X. Chen, H. Fan, and J. Qian, "Separation method of bending and torsion in shape sensing based on FBG sensors array," *Optics Express, Vol. 28, Issue 7, pp. 9367-9383*, vol. 28, no. 7, pp. 9367–9383, Mar. 2020, doi: 10.1364/OE.386738.

[31] Y. Yang, C. Hong, Z. Ahmed Abro, L. Wang, and Z. Yifan, "A new Fiber Bragg Grating sensor based circumferential strain sensor fabricated using 3D printing method," *Sens Actuators A Phys*, vol. 295, pp. 663–670, Aug. 2019, doi: 10.1016/J.SNA.2019.06.048.

[32] P. Di Palma, A. Iadicicco, and S. Campopiano, "Curvature Sensor Based on FBGs Embedded in 3D Printed Patches," *IEEE Sens J*, vol. 21, no. 16, pp. 17868–17874, Aug. 2021, doi: 10.1109/JSEN.2021.3083961.

[33] "Curvature Sensor Based on FBGs Embedded in 3D Printed Patches | IEEE Journals & Magazine | IEEE Xplore." Accessed: Aug. 07, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/9440944

[34] A. A. G. Abushagur, N. Arsad, and A. A. A. Bakar, "Cantilever beam with a single fiber bragg grating to measure temperature and transversal force simultaneously," *Sensors*, vol. 21, no. 6, pp. 1–13, Mar. 2021, doi: 10.3390/S21062002.

[35] N. V. Golovastikov, N. L. Kazanskiy, and S. N. Khonina, "Optical Fiber-Based Structural Health Monitoring: Advancements, Applications, and Integration with Artificial Intelligence for Civil and Urban Infrastructure," *Photonics 2025, Vol. 12, Page 615*, vol. 12, no. 6, p. 615, Jun. 2025, doi: 10.3390/PHOTONICS12060615.

Ensemble Learning and Socio-Economic Indicators for Forecasting South African School Performance: An Experimental Analysis

Njabulo Ndlela
Department of Computer Science
University of Zululand
KwaDlangezwa, South Africa
Email: ndlelanjabulo123@gmai.com

Sizakele Mathaba
Department of Computer Science
University of Zululand
KwaDlangezwa, South Africa
Email: SizakeleM@unizulu.ac.za

Alfredo Terzoli
Department of Computer Science
University of Zululand
KwaDlangezwa, South Africa
Email: TerzoliA@unizulu.ac.za

***Abstract*—Prediction of educational performance remains a critical challenge in South Africa, where inequalities between quintile funding and urban-rural divides restrict policy effectiveness. This study frames school performance forecasting as a supervised classification task using panel data (2012–2023) that integrates National Senior Certificate results, Department of Basic Education master lists, and municipal socio-economic indicators. We evaluated six ensemble learners, covering bagging and boosting approaches, under four policy-relevant thresholds of 'success' (50%, 60%, 70%, 80% pass rates). The results show that the predictive accuracy peaks at the cut-off point 50%, while stricter thresholds yield more balanced class distributions but harder prediction tasks. Boosting methods generally outperform bagging at mid-thresholds, though Random Forests regain strength at the strictest cutoff. Importantly, balanced accuracy (Macro-F1) remains stable across thresholds, showing that ensembles adapt even as low-performing schools in poorer quintiles and rural areas remain systematically harder to predict. This study contributes the first threshold-sensitive benchmark for school-level prediction in South Africa, highlighting both the promise and the equity risks of predictive analytics and motivating future integration of explainable AI to support fairer education policy.**

***Index Terms*—Educational Data Mining, Ensemble Learning, School Performance Prediction, Threshold Sensitivity, Explainable AI, South Africa, Policy Analytics**

I. Introduction

Education is widely recognized as a central driver of social and economic development, yet persistent disparities in quality and outcomes remain a defining challenge across much of sub-Saharan Africa. In South Africa, systemic inequalities rooted in apartheid-era legacies continue to shape school performance, with resource allocation stratified through the quintile funding system and marked differences between urban and rural contexts [1]. Recent government reports highlight that while overall access to schooling has improved, wide achievement gaps remain, particularly in disadvantaged schools where infrastructure, teacher supply, and socio-economic constraints undermine learning outcomes. Predictive analytics offers a promising avenue for addressing these disparities. Educational Data Mining (EDM) and machine learning methods can identify patterns in large-scale assessment and contextual datasets, enabling early detection of at-risk institutions and more effective targeting of interventions [2], [3]. International studies show that algorithms such as decision trees, random forests, and boosting techniques achieve accuracies exceeding 80% when predicting student or school performance [4], [5]. Within the South African context, ensemble approaches such as XGBoost and LightGBM have demonstrated competitive accuracy in predicting pass rates while integrating both academic and socio-economic indicators [6], [7]. Despite these advances, two critical gaps persist. First, most prior EDM studies either focus on higher education or international datasets, with limited evidence at the South African school level where performance variation is most acute. Second, the definition of "school success"—whether measured at a 50%, 60%, 70%, or 80% pass threshold—has not been systematically explored, even though such policy cutoffs directly influence classification outcomes and resource targeting. For example, stricter thresholds may flip the minority class from low-performing to high-performing schools, altering both predictive accuracy and equity implications. This study addresses these gaps by building and evaluating ensemble machine learning models for school-level performance prediction in South Africa. Using an integrated panel dataset (2012–2023) that combines the Department of Basic Education's National Senior Certificate (NSC) reports, the national Master List of schools, and municipal socio-economic indicators from Statistics South Africa, we investigate: (i) which algorithms provide the most reliable predictive performance across multiple definitions of success; and (ii) how model errors vary by key equity-relevant subgroups such as quintile and urban–rural location. In doing so, the paper contributes not only a technical benchmark for predictive modeling but also policy-relevant insights into the risks and opportunities of applying data-driven methods to one of South Africa's most pressing developmental challenges.

II. Related Work

Research on predicting educational performance has evolved from traditional statistical models to machine learning and ensemble approaches, each with trade-offs in accuracy, in-

terpretability, and contextual relevance. Early studies employing logistic regression and linear models offered interpretable coefficients but often failed to capture nonlinear relationships in educational data, limiting predictive power in heterogeneous contexts [6].

In contrast, tree-based and boosting methods such as Random Forest and LightGBM demonstrated higher predictive accuracy and stability while retaining some interpretability, particularly when paired with SHAP explanations [6], [8]. While linear regression continues to be valued for its simplicity and clarity, its inability to model complex, nonlinear effects makes it less suitable in dynamic educational environments compared to ensemble approaches that can adapt to heterogeneous inputs.

Comparative reviews consistently highlight decision trees and random forests as popular algorithms, with ensemble methods such as gradient boosting frequently outperforming simpler classifiers across diverse educational datasets [2], [9], [10].

For instance, accuracies between 74–87% have been reported in Portugal and Pakistan using decision trees, Naïve Bayes, and neural networks [2], [9], while ensemble approaches in Brazil and international PISA datasets improved baseline performance by up to 50% [10]. However, these contexts often benefited from richer digital data streams—such as attendance logs, forum participation, or learning management system data—that are largely absent in rural South African schools. This creates a contrast: international findings demonstrate the raw power of ensemble learning, but their portability into resource-constrained African contexts is limited by structural inequities in data infrastructure.

South African studies provide an important counterpoint by emphasizing the integration of socio-economic and community-level factors with school-level data. Wandera et al. [6] demonstrated that community attributes such as water access, electricity, household goods, and safety conditions significantly shaped pass rates. This highlights the limitations of academic-only models, which often perform well in digitally advanced settings but fail to capture structural inequalities that drive outcomes in South Africa.

Similarly, the work of [7] applied a J48 algorithm to Gauteng primary schools and reported 99.13% accuracy, illustrating that even traditional decision tree methods can perform well when local factors are carefully integrated [7]. These results stand in contrast with studies in higher education in Europe, where predictors such as digital textbook use or online course participation dominate [4], [11]. The divergence underscores that predictive success depends not only on algorithmic sophistication but also on contextual alignment between data and educational realities.

Another important distinction lies between black-box models and explainable approaches. While neural networks and support vector machines can model nonlinear boundaries effectively, they are often criticized for opacity, making them less actionable in policy contexts [8], [9]. Explainable AI (XAI) methods—such as SHAP, LIME, and counterfactuals—are increasingly applied to educational data mining to bridge this gap. Studies in South Africa have shown that SHAP values reveal how features such as sanitation, electricity, and safety contribute to predictions, enabling more transparent policy discussions [6], [8], [12].

In contrast, studies outside Africa often prioritize predictive accuracy over interpretability, reporting accuracies above 90% but offering limited insight into the mechanisms behind predictions [2], [9]. This contrast reflects different priorities: while international work often emphasizes technical performance, South African research stresses interpretability and policy relevance.

Moreover, the comparative lens across different educational levels reveals further contrasts. In South African secondary schools, models that combine socio-economic and infrastructural features outperform purely academic ones [6], while in international undergraduate settings, course-level academic performance remains the strongest predictor of outcomes [4], [9]. Even within South Africa, urban schools with better access to electricity and resources show significantly higher predicted performance than rural schools, reflecting persistent inequalities that predictive models alone cannot resolve without socio-economic awareness [6], [10]. These findings echo earlier work by Gamazo and Martínez-Abad [2], who found socioeconomic status to be the most critical predictor across 78 countries in PISA data, further reinforcing the global salience of contextual factors.

Taken together, the literature reveals a trajectory from accuracy-focused, single-model studies toward ensemble and explainable approaches that better capture the complexity of educational performance. International research demonstrates that machine learning can achieve high accuracy with detailed digital datasets, while African research stresses the necessity of integrating socio-economic and infrastructural indicators to ensure contextual validity. The convergence of these strands points toward a future where predictive modeling in education must balance technical sophistication with explainability and contextual awareness. Nevertheless, significant gaps remain in assessing how these methods generalize to resource-constrained African contexts, which this study directly addresses by evaluating ensemble machine learning on South African school performance data.

III. Data

A. Data Sources

The dataset used in this study was constructed from multiple official sources. First, the annual National Senior Certificate (NSC) school performance reports, published by the Department of Basic Education (DBE), were collected for the period 2012–2023. These reports contain school-level examination outcomes, including pass rates and related performance indicators. To enrich these outcomes with contextual information, we incorporated the national master list of schools, which provides metadata such as school identifiers, geographic location, and quintile classification. Both datasets were linked through the

unique Education Management Information System (EMIS) number, ensuring reliable integration across years.

To further capture the socio-economic environment of schools, we complemented the DBE sources with municipal-level census indicators from Statistics South Africa (Stats SA). Because the school master list includes the municipality code for each institution, census-derived socio-economic variables (e.g., household income, access to services, demographic structure) were broadcast to the school level. This integration enabled the creation of a combined dataset that reflects not only examination performance but also the broader infrastructural and socio-economic conditions in which schools operate.

TABLE I: Data sources and feature categories used in the study

Source	Features Provided
DBE NSC Reports (2012–2023)	• Examination outcomes: percentage achieved, total wrote, total achieved • Derived target: performance class (binary, based on pass-rate threshold)
DBE Master List of Schools	• School identifiers: EMIS number • School characteristics: number of learners, educators, phase, province, quintile, urban/rural, sector (public/private), type (DOE/independent), specialisation, fee-paying status, boarding school indicator
Stats SA Census Data (broadcast via municipality code)	• Demographics: population totals, sex ratio, age-specific attendance (5–24 years) • Socio-economic indicators: dwelling type, access to water, sanitation (toilet type), electricity and lighting source, refuse disposal, household education attainment • Population group composition: % Black African, Coloured, Indian/Asian, White, Other

B. Data Scope and Coverage

The integrated dataset covers the period from 2012 to 2023, yielding a longitudinal panel of South African secondary schools across all nine provinces. Each record corresponds to a school-year observation, linked through the EMIS identifier. After integration and preprocessing, the panel comprises 6,133 unique schools and 80,412 records accross the entire dataset.

The feature space spans 68 variables, grouped into academic outcomes, school characteristics, and socio-economic indicators (see Table I). These include pass rates, enrolment, educator counts, quintile classification, urban–rural location, and a rich set of census-derived indicators such as water access, sanitation, electricity, refuse disposal, household education attainment, and population group composition.

Geographically, the dataset captures the full heterogeneity of South Africa's schooling system, including urban and rural institutions as well as the socio-economic stratification reflected in the quintile system. This breadth of coverage enables not only predictive modelling but also subgroup analysis of error distribution by province, quintile, and urban–rural context.

C. Target Variable

The dependent variable in this study is a binary indicator of *school performance class*. For each school-year record, the NSC pass rate (`pct_achieved`) is compared to a predefined threshold τ. Schools with a pass rate greater than or equal to τ are labeled as `High-performing` (class = 1), while those below τ are labeled as `Low-performing` (class = 0).

To assess the sensitivity of predictive outcomes to the operational definition of "success," four thresholds were considered: 50%, 60%, 70%, and 80%. The 50% threshold corresponds to the DBE's official pass requirement, while the higher cutoffs provide stricter definitions of high performance. As the threshold increases, the class distribution shifts from a majority of high-performing schools (at 50%) to a majority of low-performing schools (at 80%).

Figure 1 illustrates this shift in class balance, showing the proportion of schools labeled as high- versus low-performing under each threshold definition.

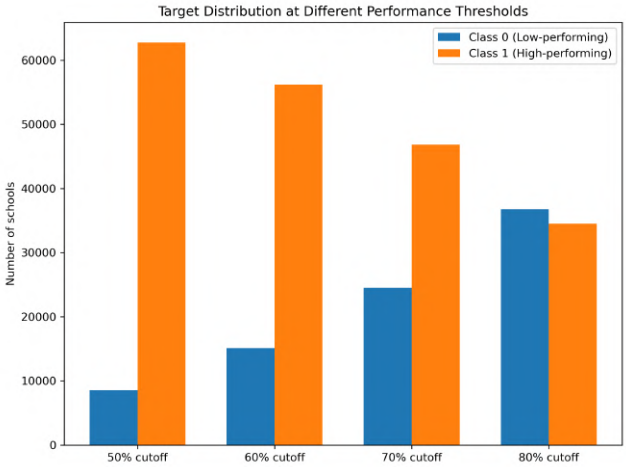


Fig. 1: Class distribution of high- and low-performing schools across different pass rate thresholds (50%, 60%, 70%, 80%).

D. Data Preprocessing

a) Record linkage and de-duplication.: NSC reports were joined to the DBE master list via the EMIS identifier; records with missing or invalid EMIS were excluded to prevent mismatches across years. When duplicated school-year entries were encountered, the most complete record was retained.

b) Data cleaning.: Implausible values were removed or clipped: for instance, school pass rates above 100% or negative counts (`total_wrote`, `ml_learners`, `ml_educators`) were treated as data errors and excluded. Province and municipality codes were validated against reference enumerations.

c) Feature encoding.: Categoricals needed to be numeric for ensemble learners. We used: (i) one-hot encoding for non-ordinal fields (e.g., `ml_phase`, `ml_province`, `ml_urban_rural`, `ml_specialisation`), which preserves distance properties for tree models; (ii) binary (0/1) encoding for dichotomous indicators (e.g., `ml_no_fee_school`, `ml_boarding_school`), which avoids needless sparsity and keeps semantics clear; and (iii) integer encoding for well-ordered categories when ordinal meaning was explicit (e.g., quintile Q1–Q5 as `quintile_raw`).

d) Missing values.: To minimize bias and retain coverage, numerical features with limited missingness were imputed using median values within province (robust to skew); categorical features were imputed with an `Unknown` level or the provincial mode. Imputation was fit on training folds only to avoid leakage.

e) Leakage control.: We dropped or refrained from constructing features with post-outcome information or trivial proxies for the label (e.g., any transformed duplicates of `pct_achieved`, engineered aggregates derived from the same target year, or features computed using test data statistics). All encoders and imputers were trained on the training split/folds only.

f) Class imbalance.: Rather than synthetic resampling, we adopted a cost-sensitive approach compatible with tree ensembles: per-sample weights equal to inverse-frequency, multiplied by a policy weight for the minority class (cost ratio set to 3.0 in our main runs). This directly penalizes minority errors during training and avoids distributional artifacts introduced by oversampling. Final decision thresholds were tuned on validation to maximize macro-F1.

g) Target definition and sensitivity.: The binary label (`performance_class`) was defined by comparing `pct_achieved` to a threshold $\tau \in \{50\%, 60\%, 70\%, 80\%\}$. Because τ alters class balance, we report models across all four settings and visualize the class shares per cutoff (Fig. 1). At 50% (DBE policy), high-performing schools dominate; balance improves around 70%; at 80% the majority flips.

h) Post-preprocessing shape.: After cleaning, encoding, and integration, the working panel comprised 71,303 school-year rows and 69 features.[1]

E. Descriptive Statistics

Table II reports the distribution of schools across the South African quintile funding system (Q1–Q5), while Table III summarizes the urban–rural split. Finally, Table IV provides summary statistics for key continuous variables (e.g., pass rate, enrolment, learner–teacher ratio). These statistics highlight the socio-economic and geographic heterogeneity in the data.

[1]Counts reflect the final dataset used for modeling.

TABLE II: Distribution of schools by quintile (Q1–Q5).

Quintile	Count	Percent
1	19,450	27.3
2	23,141	32.5
3	15,198	21.3
4	6,281	8.8
5	7,233	10.1

TABLE III: Distribution of schools by urban–rural classification.

Category	Count	Percent
Rural	53,759	75.4
Urban	17,544	24.6

TABLE IV: Descriptive statistics of key continuous variables (overall).

Variable	Mean	Q1	Median	Q3	Max
Learners	661.09	321.00	577.00	942.00	3,795
Educators	23.86	12.00	21.00	33.00	990
Pass rate (%)	74.91	63.09	78.70	90.90	100
Achieved	66.21	24.00	49.00	93.00	739
Wrote	86.36	35.00	68.00	121.00	747
Progressed learners	11.19	1.00	5.00	14.00	325

IV. Methods

A. Problem Formulation

We cast school performance prediction as a supervised binary classification task at the school–year level. For each school i and year t, the dependent variable $y_{it} \in \{0, 1\}$ denotes whether the National Senior Certificate (NSC) pass rate $\texttt{pct_achieved}_{it}$ meets or exceeds a policy threshold τ (class 1: "high-performing") or falls below it (class 0: "low-performing"). To examine how the operational definition of "success" affects predictive behavior, we evaluate $\tau \in \{50\%, 60\%, 70\%, 80\%\}$.

Let $\mathbf{x}_{it} \in \mathbb{R}^p$ denote the feature vector comprising school characteristics (e.g., enrolment, educators, phase, quintile, urban/rural) and municipal socio-economic indicators (e.g., water, sanitation, electricity, refuse disposal, household education attainment, population composition). The learning problem is to estimate a scoring function $f : \mathbb{R}^p \to [0, 1]$ that maps $\mathbf{x}_{it}$ to a calibrated probability $s_{it} = f(\mathbf{x}_{it}) = \mathbb{P}(y_{it} = 1 \mid \mathbf{x}_{it})$, which is then converted to a class label by a decision threshold θ tuned on validation data.

B. Feature Analysis

Feature importance was estimated using tree ensembles (LightGBM when available, otherwise Random Forests) on stratified training splits with balanced weights. Scores were derived from built-in importances or, where needed, permutation importance. One-hot encoded variables were aggregated back to their source columns for interpretability.

Features were ranked by normalized importance (summing to 1.0). Two schemes were tested: (i) cumulative importance

up to 95% of total weight, and (ii) Top-K with $K = 30$. Unless noted, the cumulative rule was used.

At the 50% cutoff, structural and socio-economic variables dominated. Enrolment size, educator counts, quintile, and urban–rural classification were the strongest predictors, alongside community indicators such as water, sanitation, and household education. Similar patterns held at stricter thresholds. Figure 2 shows the top 20 features by normalized importance.

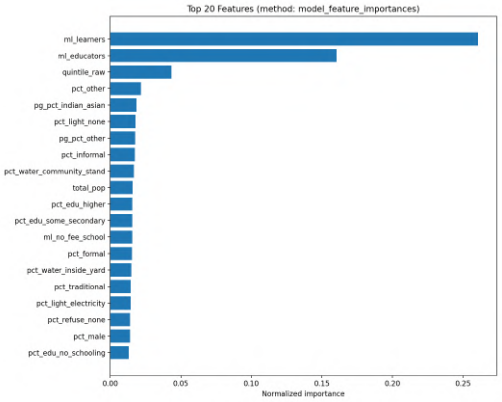


Fig. 2: Top 20 features ranked by normalized importance (50% cutoff). Enrolment size and educator counts dominate, reflecting structural resource effects, while contextual socio-economic variables (e.g., sanitation, water, household education) also contribute. Similar patterns were observed across other cutoffs.

C. Modeling Approach

This study employs a variety of learning approaches drawn-from several families. In terms of family perspective, the classifiers are summarized here. Some renowned ensemble classifiers used in this study are highlighted in Figure 3. These methods were chosen for three main reasons: (i) their ability to capture complex nonlinear relationships between school characteristics and outcomes; (ii) robustness to multicollinearity and heterogeneous feature scales; and (iii) built-in mechanisms for handling feature importance and class imbalance.

a) Bagging Models.: Random Forests and Extra Trees aggregate predictions from multiple decorrelated decision trees trained on bootstrap samples of the data. The final prediction is obtained by majority voting. This reduces overfitting and provides a strong bagging-based baseline.

b) Boosting models.: XGBoost and LightGBM extend decision trees via gradient boosting, sequentially fitting new trees to correct the errors of prior ones. For binary classification with probability scores s_{it}, the loss function minimized is the logistic loss:

$$\mathcal{L} = -\sum_{i,t} \left[y_{it} \log(s_{it}) + (1 - y_{it}) \log(1 - s_{it}) \right],$$

with regularization terms to penalize overly complex trees. LightGBM was prioritized due to its efficiency with large datasets and categorical features, while XGBoost provided a widely validated benchmark.

c) Cost-sensitive training.: To address class imbalance, particularly at the 50% cutoff where high-performing schools dominate, we applied cost-sensitive weighting. For the minority class c, weights were scaled by a factor λ such that:

$$w_c = \lambda \cdot \frac{N}{2 \cdot N_c},$$

where N_c is the number of samples in class c and N the total number of observations. This increases the penalty for misclassifying minority schools without artificially resampling the data.

d) Threshold tuning.: Although models produce calibrated probabilities s_{it}, the default classification threshold $\theta = 0.5$ may not maximize fairness across classes. We therefore tuned θ on a held-out validation set to maximize the macro-averaged F1 score:

$$\theta^* = \arg\max_{\theta} \ \mathrm{F1}_{\mathrm{macro}}(\theta).$$

This approach balances precision and recall across both low- and high-performing schools.

.

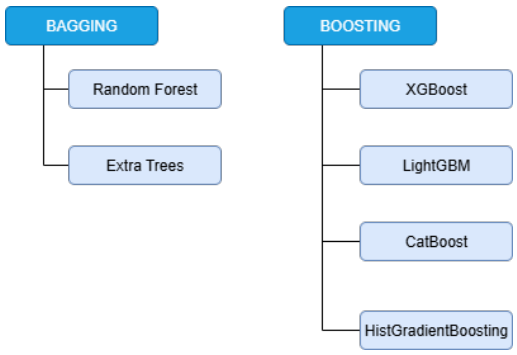


Fig. 3: Ensemble classifiers used in this study

D. Evaluation Metrics

Model performance was assessed using multiple complementary metrics to capture both overall accuracy and fairness across classes.

a) Accuracy.: Accuracy measures the proportion of correctly classified instances:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN},$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. While intuitive, accuracy can be misleading under class imbalance, as a model that always predicts the majority class may still achieve high accuracy.

b) Precision and Recall.: Precision quantifies the fraction of predicted positives that are correct, while recall (or sensitivity) measures the fraction of actual positives that are correctly predicted:

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN}.$$

These metrics trade off against one another depending on the decision threshold θ.

c) F1 score.: The harmonic mean of precision and recall balances both types of errors:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Because the data exhibit significant class imbalance, we report the *macro-averaged* F1 score, defined as the unweighted mean of class-wise F1:

$$\text{F1}_{\text{macro}} = \frac{1}{2} \left(\text{F1}_{\text{class 0}} + \text{F1}_{\text{class 1}} \right).$$

Macro-F1 ensures that performance on minority schools (low-performing class) contributes equally to the evaluation, avoiding dominance by the majority class.

d) ROC-AUC.: The area under the Receiver Operating Characteristic curve (ROC-AUC) measures the probability that the model ranks a randomly chosen positive instance higher than a randomly chosen negative instance. Unlike accuracy, ROC-AUC is threshold-independent and evaluates ranking quality across the full range of possible thresholds.

e) Metric choice.: While accuracy and ROC-AUC provide useful aggregate benchmarks, the focus of this study is on macro-F1, as it best reflects balanced predictive performance in contexts with unequal class distributions. This is particularly important in education policy, where accurately identifying low-performing schools (the minority class) is as critical as confirming high-performing ones.

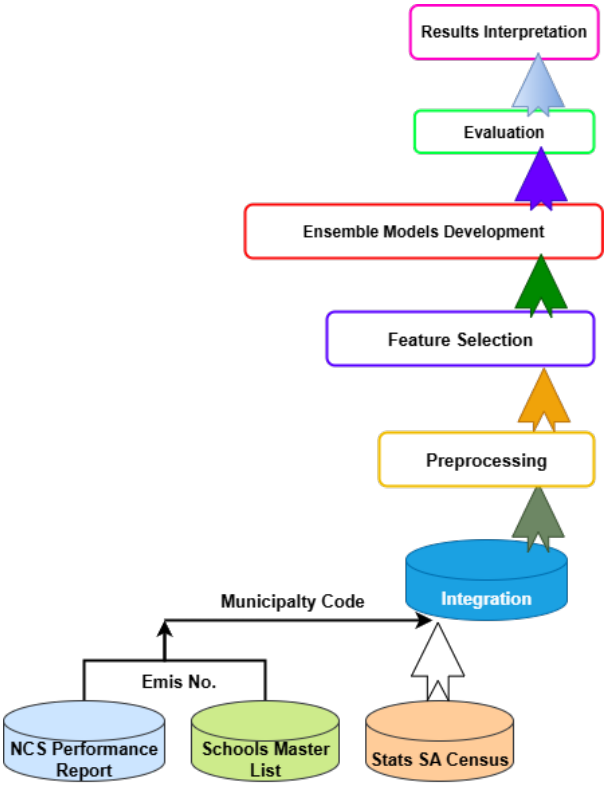


Fig. 4: High-level modeling pipeline: from integrated data sources through preprocessing, ensemble learning, threshold tuning, and evaluation.

Figure 4 presents a high-level view of our modeling pipeline. While the individual components (e.g., preprocessing, feature analysis, cost-sensitive training and more) are described in detail in Section V, the schematic highlights the sequential flow from raw data integration through to evaluation.

V. Experimental Setup

A. Data Splitting and Validation Strategy

To ensure fair evaluation and avoid information leakage, we adopted a stratified hold-out protocol at the school–year level. The integrated dataset was randomly partitioned into training (80%) and test (20%) sets, stratified by the binary target label to preserve class proportions. Within the training set, a further 15% was carved out as a validation subset used exclusively for probability threshold tuning and early stopping.

Formally, let $\mathcal{D}$ denote the full dataset, $\mathcal{D}_{\text{train}}$ the training partition, $\mathcal{D}_{\text{val}}$ the validation slice, and $\mathcal{D}_{\text{test}}$ the final hold-out set. The split was defined as:

$$\mathcal{D} \longrightarrow \begin{cases} \mathcal{D}_{\text{train}} : 68\% \\ \mathcal{D}_{\text{val}} : 12\% \\ \mathcal{D}_{\text{test}} : 20\% \end{cases}$$

where proportions are relative to the full dataset. All splits were stratified by $y \in \{0, 1\}$ to mitigate class imbalance effects across partitions.

The validation set served two purposes: (i) tuning the decision threshold θ to maximize macro-F1, and (ii) monitoring performance for early stopping in gradient boosting models. The test set remained untouched until final evaluation. This protocol was repeated across all four performance thresholds ($\tau \in \{50\%, 60\%, 70\%, 80\%\}$) to assess sensitivity to the definition of "high-performing" schools.

B. Hyperparameter Tuning

Hyperparameters were optimized separately for each ensemble learner to balance predictive accuracy, robustness, and computational efficiency. We used a grid search on the training set with 5-fold stratified cross-validation, selecting configurations that maximized macro-F1. Early stopping on the validation subset prevented overfitting in boosting models.

For gradient boosting models, tuning focused on controlling tree depth, learning rate, and the number of boosting rounds. For Random Forests, emphasis was on the number of trees and maximum depth. Cost-sensitive weighting was integrated throughout, with the class cost ratio $\lambda \in \{1.5, 2.0, 3.0\}$ tested to account for imbalance.

Table V summarizes the main hyperparameters and search ranges considered.

TABLE V: Hyperparameter search ranges for ensemble learners.

Model	Hyperparameters	Search Range
Random Forest	$n_estimators$	$\{200, 400, 600, 800\}$
	max_depth	$\{6, 10, 15, None\}$
	$class_weight$	$\{balanced, balanced_subsample\}$
XGBoost	$n_estimators$	$\{200, 400, 800\}$
	max_depth	$\{4, 6, 8\}$
	$learning_rate$	$\{0.01, 0.05, 0.1\}$
	$subsample$	$\{0.8, 1.0\}$
	$colsample_bytree$	$\{0.8, 1.0\}$
	$scale_pos_weight$	$\{1.5, 2.0, 3.0\}$
LightGBM	$n_estimators$	$\{500, 1000, 1500\}$
	$learning_rate$	$\{0.01, 0.03, 0.05\}$
	max_depth	$\{-1, 6, 10\}$
	num_leaves	$\{31, 63\}$
	$subsample$	$\{0.8, 1.0\}$
	$colsample_bytree$	$\{0.8, 1.0\}$
	$class_weight$	$\{balanced\}$

The final models typically converged to medium-depth trees ($max_depth \approx 6$–8) and moderate learning rates ($\eta \approx 0.03$–0.05). These settings provided stable generalization while retaining sensitivity to minority schools.

C. Training Procedure

Once hyperparameters were selected, final models were retrained on the combined training and validation partitions, with the untouched test set reserved for evaluation. The training protocol incorporated the following elements:

a) Early stopping.: For gradient boosting models (XGBoost, LightGBM, CatBoost), training was monitored on the validation subset using the macro-F1 and average precision metrics. Training was halted if no improvement was observed for 200 consecutive rounds, and the best iteration was retained. This avoided overfitting and reduced computational cost.

b) Cost-sensitive weighting.: To mitigate class imbalance without resampling, we employed per-sample weights. For the minority class c, the weight was scaled by a factor λ, as described earlier:

$$w_c = \lambda \cdot \frac{N}{2 \cdot N_c},$$

where N_c is the number of samples in class c and N the total number of observations. Empirically, $\lambda \in \{2.0, 3.0\}$ produced the most balanced results across metrics.

c) Threshold tuning.: Ensemble models output calibrated probabilities $s_{it} \in [0, 1]$. Rather than defaulting to $\theta = 0.5$, we tuned the classification threshold on the validation set to maximize macro-F1:

$$\theta^* = \arg\max_{\theta \in [0,1]} \text{F1}_{\text{macro}}(\theta).$$

The chosen θ^* was then applied to the test set, ensuring fair performance across both low- and high-performing schools.

d) Reproducibility.: All experiments were seeded with a fixed random state (42) to ensure reproducibility. Models were trained on the same preprocessed feature set produced by the feature analysis pipeline, ensuring consistency across cutoffs. This protocol was repeated independently for each performance threshold ($50\%, 60\%, 70\%, 80\%$), enabling direct comparison of how model behavior changes with different definitions of school success.

VI. Results

A. Model Comparison

We first compare the predictive performance of six ensemble learners across four threshold definitions of school success ($\tau \in \{50\%, 60\%, 70\%, 80\%\}$). Tables VI and VII report test Accuracy, Macro-F1, and ROC-AUC, while Figures 5 and 6 summarize results visually.

TABLE VI: Model comparison at 50% and 60% cutoffs (best per column in **bold**).

2*Model	**50% cutoff**			**60% cutoff**		
	Acc	Macro-F1	ROC-AUC	Acc	Macro-F1	ROC-AUC
XGBoost	0.819	**0.744**	**0.806**	0.758	0.773	**0.793**
LightGBM	0.823	0.740	0.801	0.767	**0.777**	0.792
CatBoost	0.806	0.740	0.804	0.760	0.772	0.792
HistGradBoost	0.828	0.739	0.801	**0.769**	0.767	0.789
Random Forest	**0.847**	0.713	0.768	0.768	0.740	0.761
Extra Trees	0.833	0.713	0.729	0.764	0.736	0.738

TABLE VII: Model comparison at 70% and 80% cutoffs (best per column in **bold**).

2*Model	**70% cutoff**			**80% cutoff**		
	Acc	Macro-F1	ROC-AUC	Acc	Macro-F1	ROC-AUC
XGBoost	0.709	0.785	0.768	0.664	0.758	**0.773**
LightGBM	0.711	**0.786**	0.770	0.662	0.755	0.771
CatBoost	0.703	0.784	0.768	0.652	0.742	0.771
HistGradBoost	0.699	0.777	0.760	0.644	0.731	0.769
Random Forest	0.692	0.765	0.744	**0.681**	**0.778**	0.745
Extra Trees	0.681	0.755	0.722	0.671	0.770	0.729

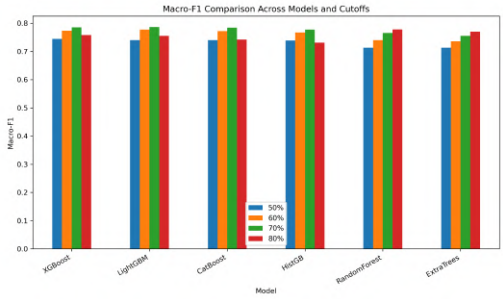


Fig. 5: Macro-F1 comparison across models and cutoffs. Boosting models dominate at mid-thresholds, while bagging methods (Random Forest, Extra Trees) improve at 80%.

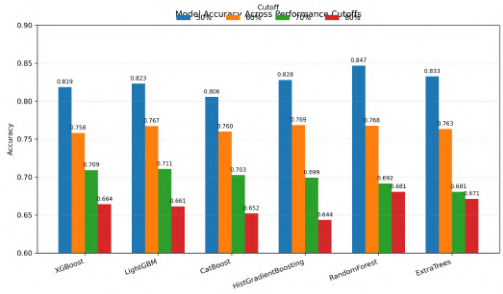


Fig. 6: Accuracy comparison across models at different cutoffs. At $\tau = 50\%$, accuracy peaks above 0.84 with Random Forest, while stricter cutoffs reduce absolute accuracy for all learners.

B. Threshold Analysis

To complement performance scores, we examine how predictive calibration and threshold sensitivity behave under the 50% cutoff, which in our experiments produced the strongest overall balance between accuracy and Macro-F1. At this threshold, the two classes (low- vs. high-performing schools) are more reasonably represented, giving the models a more balanced signal to learn from. This contrasts with stricter cutoffs (60%, 70%, 80%), where the class distribution becomes more skewed and prediction of the minority group is harder.

Figure 7 shows reliability curves for all six models at the 50% cutoff. The diagonal represents perfect calibration: points above the line indicate under-confidence, while points below the line indicate over-confidence.

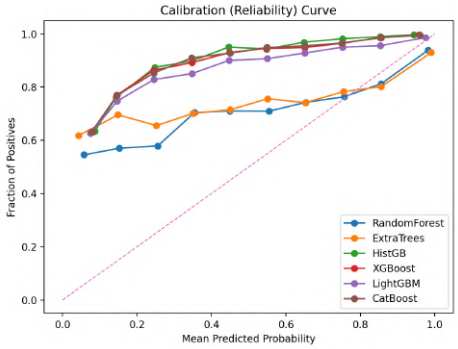


Fig. 7: Calibration (reliability) curves at the 50% cutoff. The diagonal line indicates perfect calibration.

Boosting approaches (XGBoost, LightGBM, CatBoost, and HistGradientBoosting) remain closer to the diagonal, indicating better-calibrated probability estimates compared to bagging models (Random Forest, Extra Trees). This suggests that while bagging achieves competitive accuracy, boosting methods yield probability scores that are more trustworthy for downstream decision-making.

We also conducted a threshold sweep, varying the decision threshold θ between 0.1 and 0.9. Figure 8 shows that boosting models maintain superior Macro-F1 stability across operating points, confirming robustness at the 50% cutoff where class balance is least distorted.

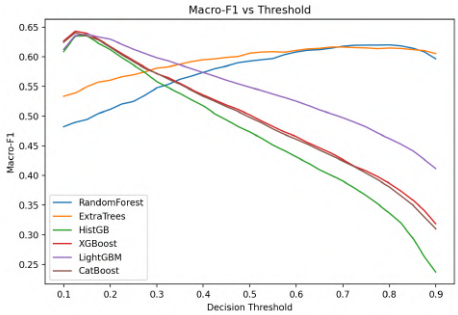


Fig. 8: Threshold sweep of Macro-F1 scores at the 50% cutoff. Boosting methods sustain higher Macro-F1 across thresholds, reflecting stronger balance between precision and recall.

C. Error Analysis

While aggregate performance metrics provide a global view of accuracy, it is equally important to investigate *where* models fail. We therefore analyze misclassifications by socio-economic subgroups to reveal potential sources of systematic bias.

1) Errors by Quintile: Figure 9 reports the distribution of errors across school quintiles at the 50% cutoff. Low-quintile schools (Quintile 1–2), which represent the poorest institutions, exhibit disproportionately high misclassification rates, with the model frequently failing to identify them as low-performing. By contrast, Quintile 5 schools show the lowest

error rates, suggesting that performance prediction is more stable in affluent contexts.

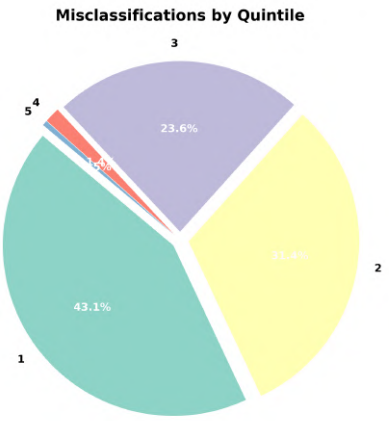


Fig. 9: Misclassification rates by school quintile (50% cutoff).

2) Errors by Urban–Rural: Figure 10 compares misclassification rates between urban and rural schools. Rural schools are more likely to be misclassified, reflecting the challenges posed by sparse infrastructure and weaker socio-economic conditions. This finding underscores the need for context-sensitive modeling approaches and cautions against one-size-fits-all deployment.

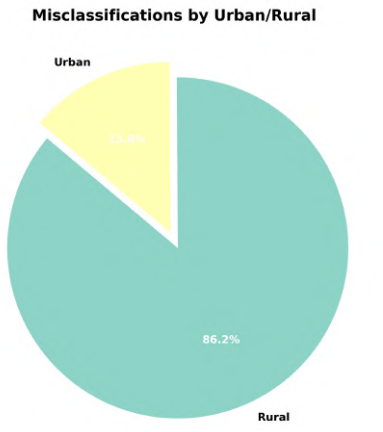


Fig. 10: Misclassification breakdown by urban vs. rural schools at the 50% cutoff.

VII. Discussion

A. Interpretation

The results highlight three key patterns. First, ensemble boosting methods (XGBoost, LightGBM, CatBoost) consistently outperformed bagging approaches across most thresholds, confirming their strength on heterogeneous tabular features with nonlinear interactions [13]–[15]. Random Forest regained relative strength at the strict 80% cutoff, suggesting bagging methods may be less sensitive to extreme class imbalance.

Second, predictive accuracy peaked at the 50% cutoff, where class distributions are most balanced. At stricter thresholds (70–80%), the task became harder as one class became increasingly rare. Nonetheless, Macro-F1 remained relatively stable across cutoffs, indicating that ensemble learners adapt to shifting definitions of "success" without collapsing on minority classes.

Third, low-performing schools remained consistently harder to classify across all thresholds. This reflects both statistical imbalance—fewer informative examples for the minority class—and structural variability in disadvantaged contexts, where infrastructure, household SES, and teacher supply create noisier feature–outcome relationships [1]. In short, "hard cases" cluster in settings where features are weaker proxies for year-to-year performance.

Overall, these patterns show that model performance is sensitive to how "success" is defined, with best results at 50% and increasing difficulty at stricter thresholds. This underscores the importance of evaluating models across multiple cutoffs and focusing not only on accuracy but also on class-wise performance.

B. Limitations

Several limitations must be acknowledged. First, the integrated dataset—though large and longitudinal—remains constrained by data quality: schools with missing identifiers were excluded, and socio-economic indicators were only available at municipal level, risking ecological fallacy in heterogeneous districts [1].

Second, class imbalance persisted across thresholds. Even with cost-sensitive training, the minority class (low-performing schools at $\tau = 50\%$ or high-performing at $\tau = 80\%$) remained harder to classify, reflecting both statistical skew and the structural volatility of disadvantaged schools.

Third, the models capture retrospective associations rather than causal mechanisms. Policy shifts or shocks—such as curriculum reforms or COVID-19—may alter feature–outcome relationships in ways not reflected in historical data [2].

Finally, ensemble methods, while accurate, remain opaque. Incorporating explainable AI (e.g., SHAP, LIME, counterfactuals) is essential to translate predictions into insights that policymakers can trust [8], [12].

Overall, these limitations call for cautious interpretation and highlight that predictive accuracy alone is insufficient without contextual validation and interpretability.

C. Future Work

Future work should extend this study in several directions. First, incorporating explainable AI methods (e.g., SHAP, LIME, counterfactuals) can clarify which features drive misclassification and improve interpretability [8], [12]. Second, testing transferability across provinces and years would show whether models trained in one context generalize to others or require local calibration. Third, adding behavioral and

attendance data could enhance prediction, as prior studies highlight these as strong early-warning indicators [2], [9]. Finally, embedding models into dynamic dashboards that update with new NSC and census data could translate prediction into practical monitoring tools.

In short, future research should focus on making models not only accurate but also interpretable, transferable, and operational.

VIII. Conclusion

This paper demonstrated the feasibility of predicting school performance in South Africa using ensemble methods applied to integrated NSC, school, and municipal data (2012–2023). Boosting algorithms (XGBoost, LightGBM, CatBoost) consistently outperformed bagging at mid-thresholds, while Random Forest regained strength under the strictest definition of success, echoing international benchmarks that identify boosting as state-of-the-art for tabular data [13]–[15].

Persistent challenges remain: low-performing schools are systematically harder to classify due to imbalance and contextual volatility, and threshold choice alters which schools are flagged, underscoring that model performance is policy-dependent [1], [8]. Key features such as quintile, water access, and household education levels confirm that outcomes are embedded in community conditions [2], suggesting predictive tools could complement South Africa's pro-poor funding frameworks through early-warning systems.

Limitations include data quality constraints, aggregation of socio-economic indicators, and the retrospective scope. Future work should integrate learner-level and behavioral data, apply explainable AI for interpretability, and test transferability across provinces and years.

In summary, ensemble learning offers a promising foundation for school performance prediction, provided future efforts embed fairness and interpretability to ensure predictive analytics serves as a trusted decision-support tool for equitable education.

References

[1] N. Spaull, "South Africa's Education Crisis: The quality of education in South Africa 1994-2011," 2013. [Online]. Available: https://www.semanticscholar.org/paper/South-Africa's-Education-Crisis%3A-The-quality-of-in-Spaull/6e54309140a56d70af00335b5fdc4c4452235430

[2] A. Gamazo and F. Martínez-Abad, "An exploration of factors linked to academic performance in pisa 2018 through data mining techniques," *Frontiers in Psychology*, vol. 11, pp. 1–11, 2020.

[3] R. Asif, A. Merceron, S. A. Ali, and N. G. Haider, "Analyzing undergraduate students' performance using educational data mining," *Computers Education*, vol. 113, pp. 177–194, 10 2017.

[4] A. Mueen, B. Zafar, and U. Manzoor, "Modeling and predicting students' academic performance using data mining techniques," *International Journal of Modern Education and Computer Science*, vol. 8, pp. 36–42, 11 2016.

[5] S. S. Athani, S. A. Kodli, M. N. Banavasi, and P. S. Hiremath, "Student performance predictor using multiclass support vector classification algorithm," IEEE Xplore, p. 341–346, 07 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8305866

[6] H. Wandera, V. Marivate, and M. D. Sengeh, "Predicting national school performance for policy making in south africa," in *Proc. 6th Int. Conf. on Soft Computing & Machine Intelligence (ISCMI)*. IEEE, 2019, pp. 23–29.

[7] K. I. M. Ramaphosa, T. Zuva, and R. Kwuimi, "Educational Data Mining to Improve Learner Performance in Gauteng Primary Schools," in *2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, Aug. 2018, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8465478

[8] H. Wandera, "Comparing school outcomes in south africa and sierra leone using interpretable machine learning," *Education and Information Technologies*, vol. 25, no. 4, pp. 3567–3584, 2020.

[9] R. Asif, A. Merceron, S. A. Khan, and A. Pathan, "Predicting student academic performance at higher secondary level using data mining," *Computers & Education*, vol. 113, pp. 76–93, 2017.

[10] I. V. da Silva and M. T. da Silva, "Use of data mining to identify the technological resources that contribute to school performance in large-scale evaluations of brazilian high school," in *Proc. Int. Conf. on Educational Data Mining (EDM)*, 2021, pp. 451–456.

[11] P. Cortez and A. Silva, "Using data mining to predict secondary school student performance," in *Proc. 5th FUture Business Technology Conf. (FUBUTEC)*, 2008, pp. 5–12.

[12] L. C. Nnadi, Y. Watanobe, M. M. Rahman, and A. M. John-Otumu, "Prediction of students' adaptability using explainable ai in educational machine learning models," *Applied Sciences*, vol. 14, pp. 5141–5141, 06 2024.

[13] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, vol. 1, p. 785–794, 08 2016.

[14] G. Ke, Q. Meng, T. Finley, T. Wang, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 12 2017. [Online]. Available: https://www.researchgate.net/publication/378480234_LightGBM_A_Highly_Efficient_Gradient_Boosting_Decision_Tree

[15] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on tabular data?" arXiv.org, 07 2022. [Online]. Available: https://arxiv.org/abs/2207.08815

Polarization-based Optical Fibre Sensor for Landslide Monitoring

Sisipho Dlakiya
Dept. of Geoscience
Nelson Mandela University
Gqeberha, South Africa
s217683762@mandela.ac.za

F.J. Swanepoel
Dept. of Geological Science
University of Cape Town
Cape Town, South Africa
SWNFRA010@myuct.ac.za

James Jena
Centre for Broadband Communication
Nelson Mandela University
Gqeberha, South Africa
James.Jena@mandela.ac.za

Gaathier Mahed
Dept. of Geoscience
Nelson Mandela University
Gqeberha, South Africa
Gaathier.Mahed@mandela.ac.za

David Waswa
Centre for Broadband Communication
Nelson Mandela University
Gqeberha, South Africa
David.Waswa@mandela.ac.za

***Abstract*— We present a polarization-based optical fibre sensor (OFS) for detecting vibrations relating to strain-induced processes that lead to slope failure. The high sensitivity of the polarization OFS to small signals enables the detection of low-amplitude vibrations along its length for hillslope monitoring. The system was experimentally tested within the 50 Hz frequency range to monitor internal changes in the soil structure in correspondence to the ambient vibrations from environmental background noise. This was achieved by embedding the optical fibre cable in a rain-simulated, hillslope failure model that allows inclination at 15°, 30°, and 45°. The failure model was monitored through timing events on polarization fluctuations traces and spectral frequency analysis. There was no observable slope failure on the 15° incline; however, close to 640 s, the sensor detected less significant failure events related to erosion caused by oversaturation, leading to cracks that connected to form streams. In the 100% medium grain sand soil used in all experiments, saturation occurred between 250 s and 300 s as confirmed by conductive electronic moisture sensor measurements. The first observed slope failure initiated from the base of the 45° slope as time approached 525 s, followed by the 30° incline, whose slope failed just before 770 s. The delay between saturation and failure at 30° was approximately double that measured for the 45° slope. Failure precursor signatures show amplitude variation features characteristic of soil creep, desaturation, and cracking. The results demonstrate the effectiveness of optical fibre strain sensors as an early warning detection system for slope failure and landslide monitoring.**

Keywords*— optical fibre, polarization, strain sensor, slope failure, landslide monitoring.

I. Introduction

Slope failures and landslides pose significant risks to infrastructure and human life. Early detection of slope instability is crucial for implementing timely mitigation measures. Traditional slope monitoring techniques often rely on complex and costly instrumentation, limiting their widespread deployment. Optical fibre sensors (OFS) are capable of detecting strain-induced mass movements that lead to slope failure. The detected fluctuations result from shifts in the polarization state of light corresponding to the strain caused by disturbances acting on the slope. Strain occurs when external forces (e.g., temperature and pressure) change the shape of an object [1]. In the context of slope failure or landslides, strain occurs when the weight of water exceeds that of the soil [2]. Factors such as earthquakes, human activities, and heavy rainfall also result in strain. Earthquakes cause ground vibrations, which can initiate shear stresses on slopes. If the slope is unstable, the shear stress can lead to failure. Human activities such as construction and deforestation remove vegetation and can increase (construction) or reduce (deforestation) the weight of the soil. Heavy rainfall increases the soil's weight, contributing to adhesion, which initiates a slope failure [3] [4]. A slope deforms when subjected to stress; small and large amounts of stress can lead to failure, especially in steep slopes. Steep slopes are susceptible to failure and landslides because they are subjected to greater shear stresses than gentle slopes [3].

The effects of slope failure or landslides have led to the development of traditional monitoring systems. Such systems include geotechnical sensors (extensometers, piezometers, and inclinometers), slope-retaining materials, steel bars, visual observations, and geophones [5]. While these systems provide valuable data, they are often costly, point-based, and often fail to capture subtle soil movements before failure in real-time [4]. This gap requires an advanced, distributed monitoring system, such as fibre optics, which can detect both localized and distributed strain events in real-time through related polarization shifts, or changes in wavelength or phase of the optical wave within the fibre [4]. The advantage of optic fibre sensors is their ability to capture subtle, high spatial resolution data over extensive areas, enabling it to detect minute changes in the environment [6]. This ability has proven to be crucial for early warning systems and hazard mitigation [7]. It can identify preliminary failures before significant collapses [7]. Optical fibres can be used to monitor landslides because they are suitable for both small and large-scale use and are cost-effective [8]. The preference for polarization optical fibre strain sensors is due to their feasibility for real-time monitoring systems and sensitivity [8].

The polarization-based optical fibre sensor is a hybrid phase detection system with sensitivity enhanced through a

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

Michelson Interferometer. Any perturbation along the fibre length is detected through relative shifts in the state of polarization with respect to the reference phase from the steady auxiliary arm. In distributed fibre sensing, techniques are incorporated to identify the location of the disturbance. References [6] and [7] noted that DAS's detection mechanism, application range, operational frequency, and measurement focus surpass those of existing optical fibre-based approaches.

FOS uses acoustic vibrations to monitor phase changes in scattered light [6]. Brillouin- and Rayleigh-based optical fibre sensors, commonly used for strain and temperature measurements, may not capture rapid changes effectively [9]. While optical fibre sensors have been successfully applied for dynamic event monitoring, such as landslides, making them suitable for immediate hazard detection, traditional fibre optic systems are primarily designed for long-term condition monitoring, which limits their applicability for real-time slope failure detection [6] [7]. FOS can detect rapid strain and vibrations at acoustic frequencies, whereas traditional fibre sensors—like Brillouin, Rayleigh, or FBG-based systems—are slower and better suited for static or slowly varying measurements such as temperature and long-term strain.,[6] [7]. FOS measures dynamic events like landslides and utilizes the entire fibre length as sensors, resulting in continuous monitoring along the fibre. Traditional optic fibres may rely on discrete sensors placed at specific points/intervals, limiting spatial resolution and monitoring coverage [9].

The current state of the art in landslide monitoring uses various monitoring techniques which include remote sensing, geo-techniques, geodetics, geophysics, hydrological, and mapping methods [10] [11]. Remote sensing (RS) has been used since the launch of the first Landsat satellite in 1972. RS uses a satellite or an aircraft to detect and monitor the physical features of an area by measuring the reflected and emitted radiation from a distance. It features active sensors (SAR and Lidar) that generate their own signals, which are calculated when reflected from the Earth's surface. Passive sensors (Multispectral/Hyperspectral) operate across a wide range of spatial resolutions on satellites (e.g., Landsat, Sentinel, QuickBird). Unmanned Aerial Vehicles (UAVs)are widely used due to their flexibility in high-risk situations and inaccessible areas [10]. Geotechnics utilizes inclinometers, extensometers, tiltmeters, and geophones to monitor slope displacement, sliding surfaces, and slope rates. These instruments can be installed in boreholes, on surface deformations, and on the surfaces of potential landslides to provide inputs for landslide modeling and stability analysis [11]. Geodetic methods are used to track surface changes using total station surveys (GNSS) and scanning. GNSS is preferred for its flexibility, reliability, and cost-effectiveness, as it provides real-time data. In contrast, Terrestrial Laser Scanning (TLS) and Automated TLS offer 3D mapping of the Earth's surface displacement and geometry [10] [11]. Photogrammetry is commonly used for its high-resolution surface data [11]. Water is the primary trigger for slope instability under hydrogeological methods. Hydrogeology measures/monitors water-related parameters like precipitation, groundwater levels, pore water pressure, snow cover thickness, soil moisture, and water discharge. This hydrogeological data provides real-time data for potential landslide activities when collected automatically. The traditional mapping method is still fundamental for landslide studies, which require geological, geomorphological, and engineering geological data. To enhance mapping accuracy, the modern approaches have integrated GIS, Satellite Imagery, and High-resolution Digital Elevation Models (DEMs) with mapping.

The following objectives were set for this study: Firstly, to monitor the behavior of soil when the system is inclined at different angles, and secondly, to evaluate the effectiveness of a polarization-based optical fibre sensor in monitoring the frequencies and strain applied before, during, and after a slope failure event. The system demonstrates an effective and reliable technique for detecting precursors of slope instability. This study contributes to the advancement of slope stability monitoring technologies, offering an innovative solution to improve data accuracy and real-time monitoring, while enhancing the understanding and prediction of geologic hazards.

II. Principle Theory

The polarization-based optical fibre sensor used in the study operates as an anomaly detector of events within the surroundings of the entire fibre length which does not measure the location of such activity. Zoning of disturbances in distributed acoustic sensors relies on the optical back reflection and the time-of-flight technique. This technique works when an optical pulse is launched into the fibre under test and is backscattered through Rayleigh scattering, and changes in the optical field are used to detect disturbances in the form of corresponding polarization, phase, spectral or shifts [8]. Reference number [5] suggests that backscattered light can be polarized in different directions. Changes detected from the back-scattered light are induced by irregularities or disturbances (acoustic vibrations) along the cable [3]. Soil strain, vibrations, or temperature variations are some of the disturbances detected by the fibre, which alter the polarization state of the backscattered light. [8] [12].

They are monitored and analyzed simultaneously, making the optical fibre highly effective for the monitoring of geo-hazards, [13]. Soil movement caused by strain is detected by the optical fibre embedded in the soil. When an optical fibre sensor experiences disturbance in the form of strain, it stretches, thus resulting in a periodic index variation inside the core. Optical fibres are primarily designed to transmit high-quality data signals for long distances [13]; therefore, they have the dual role of detection and signal transmission in sensing. Compared to other sensing methods, the uniqueness of this technique is the ability to perform distributed sensing along the entire fibre length, offering high spatial resolution and resistance to environmental noise. These characteristics make it a promising solution for reliable and cost-effective slope monitoring applications.

Photoelastic effect is an optical technique used to approximate the relationship between applied strain (ε) and polarization shift ($\Delta\,\theta$):

$$\Delta\theta \approx \kappa \cdot \varepsilon \quad (1)$$

Where κ is the sensitivity coefficient determined by the fibre's geometry, material composition, and wavelength of operation [12].

The relationship between detected polarization shifts and strain strength is crucial for enhancing strain measurement accuracy. The multiple sensors distributed along the fibre are all affected by the local strain applied to the fibre. The strain applied to the fibre changes the polarization state of the light traveling through the fibre. This change implies that the refractive index of the fibre material is altered, thus affecting light propagation. High strain levels result in significant shifts in the polarization state. Polarization-based optical fibres then become crucial in this regard, as they monitor the polarization of light, which can enhance the sensitivity and accuracy of strain measurement. During monitoring, the sensors can detect minute changes in the polarization state of the fibre, [12]. This capability is crucial for early warning systems for potential hazards, such as ground movements or structural deformation. Every point along the length of the fibre acts as a continuous independent strain detector. Polarization OFS captures both local micro-strain and distributed slope-scale deformations, which enhances the detection of precursory landslide signals in real time. This is a unique function that traditional point sensors lack [12][14][15].

III. Methodology and Experimental Setup

A. Polarization-based Optical Fibre Sensor

A laser was used to send a light wave into the fibre, which was split into two beams by a beam splitter. The first beam was injected into a reference arm of the optical fibre on a steady optical bench, maintaining its polarization. The second beam was directed to the sensing arm of the fibre buried in the soil.

A custom optical fibre was buried in the soil to measure strain, across the lower, middle and upper slope as shown in Fig. 1. As the soil structure began to change due to external forces, the fibre sensor detected these movements as events related to variation in strain over time. Due to events in the soil, the detected polarization shifts were measured in terms of amplitude variations, which would respectively indicate the strength and frequency of the strain detected. The experiments conducted in the laboratory were inclined at different angles and had the same soil composition. Experiment 1 was inclined at 45° with 100% sand, and experiment 3 was inclined at 30 ° with 100% sand, while experiment 2 was inclined at 15 ° with 100% sand. This was done to investigate the effectiveness of the fibre sensor in monitoring various slope failure precursors and landslide events under different conditions.

B. Hillslope failure model

The hillslope failure model developed for this study allowed the inclination of the slope at 15°, 30°, and 45°, Fig.2. These angles were selected because they represent common natural slope gradients: steep mountainous terrains (>40°), moderate hillslopes (20-30°), and gentle agricultural slopes (<20°). A system with a connected 20 L reservoir, tap, and 20 L regulator was used to induce rainfall on the soil using sprinklers. This system was placed above the model, and the optical fibre was embedded in the soil. The model in Fig. 2A was built around a plastic tub with dimensions of 50cm x 32 cm x 12 cm. There were nine sprinklers, each with an average flow rate of approximately four liters of water per hour.

The G655 single-mode fibre used has a core diameter of 9 µm and a cladding diameter of 125µm. The laser source operated at a wavelength of 1550 nm with an optical power output of 4.95 dBm. The sensor recorded disturbances throughout the experiment. The tap was opened 2 minutes after the start to allow the fibre to establish a steady baseline from ambient noise. Fig. 2B shows the rainfall simulation system, which was used with tap water in all experiments, along with a bucket designed to drain and collect water from the model. The optic fibre sensor is 5m in length, 2m was buried under the same amount of the dry medium-grained sand soil, while 3m connected from the exit top side of the slope model to the splitter and displayed as shown in Fig. 1. The length of the fibre used does not significantly influence the sensitivity of the system for both the model and natural slope lengths. This is because the slope length does not span a long distance, which prevents it from affecting back reflections. Fibre back reflections can be monitored up to 25km without problems.

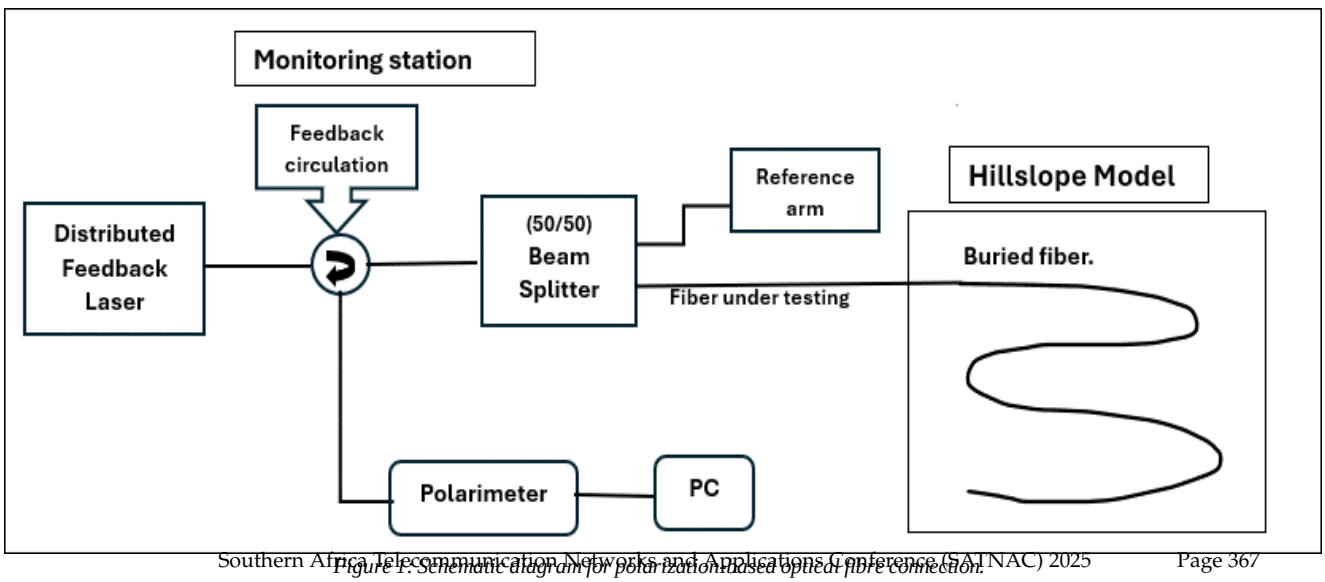


Figure 1. Schematic diagram for polarization-based optical fibre connection.

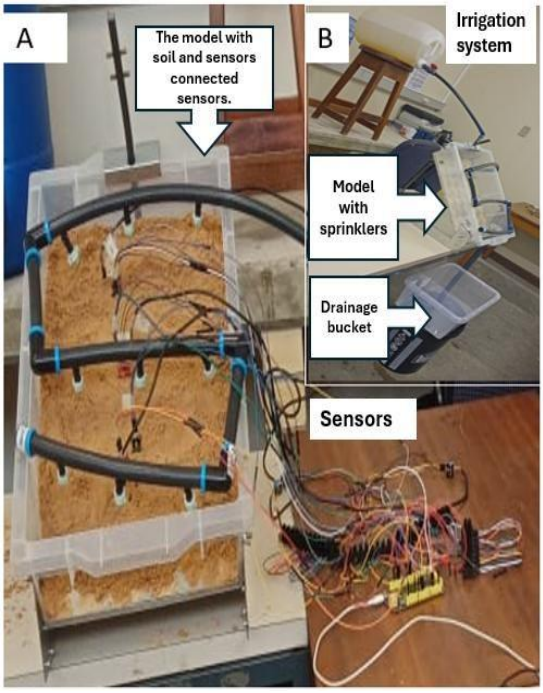


Figure 2: Hillslope monitoring model. A- is the model inclined with the soil. B- is the entire system for rainfall simulation, from the water source to the drainage.

IV. Results and Discussion

Three experiments were conducted (Fig. 3 and Fig. 4), and their results were analyzed from time-domain graphs and spectral intensity distribution curves. The experiment was stopped once the slope reached equilibrium, at which point little or no soil remained for the rainfall to erode. The series of events included infiltration of the water, followed by saturation. Surface runoff was observed when the slope reached maximum saturation, leading to finer sediment erosion. During this time, some slopes started forming small cracks which, connected to form tributaries and gullies as the rate of erosion increased, and these effects were observed on the time traces before slope failure could occur.

Fig. 3 and Fig. 4 show time domain signals of interest, while Fig. 5,6, and 7 show the intensity spectrum curves of these signals for the same composition (Sand 100%), inclined at 45°,15°, and 30°, respectively. A composition of sand inclined at 45° shows the difference in frequency of the top, middle, and bottom rows for experiment 1, in Fig. 3, representing signatures of events before slope failure (B), during slope failure (D), and after slope failure (A) at different times through the experiment. Moving from top to bottom of each column are selected strain signatures from continuously logged 2-second recordings of events within each period, shown by the sections in columns sequenced from left to right. The top, middle, and bottom rows for all experiments represent detected amplitude variations of interest captured before, during, and after failure events. This shows the difference in the amount of strain exerted on the fibre. There was a series of failure events due to the steepness of the slope. Although no major slope failure was observed for the gentle incline at 15 ° (experiment 1), minor failure events were still recorded (Figs. 3 and 4), indicating the system's sensitivity to subtle intrinsic events that are not visible from the surface.

The most significant precursors and failures are depicted in Figs. 3 and 4. Both experiments 1 and 2 had failures that preceded them. After the slope reached saturation in close to 300 seconds, the erosion of fine sediment at the bottom of the slope led to the formation of cracks in the same region. The increase in the rate of erosion lengthened the cracks, leading to slope failures (525 s, bottom row and middle row, 630 seconds), as observed in Figure 3, Experiment 1.

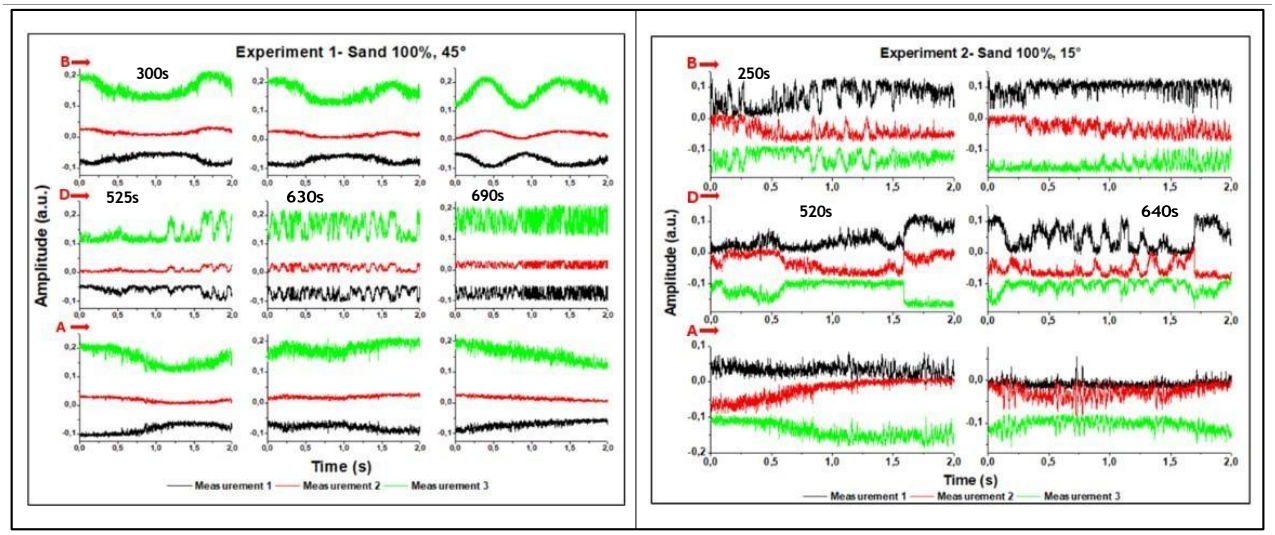


Figure 3: Time series signatures of Sand 100%, inclined at 45° (Experiment 1) and at 15° (Experiment 2). Moving from Top to Bottom of the column: Top row (B)- Before failure, Middle row (D)- During failure, and Bottom row (A)- After failure, for a couple of failure events. Measurement 1, Measurement 2, and Measurement 3 are Stokes Parameters simultaneously recorded from 3 orthogonal axes.

The failure lasted for a few seconds while the cracks moved vertically up towards the top of the slope. This led to another event of failure, which occurred 690 seconds later. After every failure event, another cycle of infiltration occurred, and the soil became saturated again. The bottom row shows the period of infiltration and saturation building up to another failure event until all the soil was washed away. The angle of inclination influenced the time required for the entire slope to fail. Saturated soil on steeper slopes failed more quickly than on gentler slopes. For example, the 30° incline exhibited a slope-failure time of 770 s after initial saturation, which is nearly twice as long as the saturation-to-failure duration observed for the 45° slope. Saturation occurs simultaneously in all stages because the rain simulator is designed to sprinkle evenly across the slope.

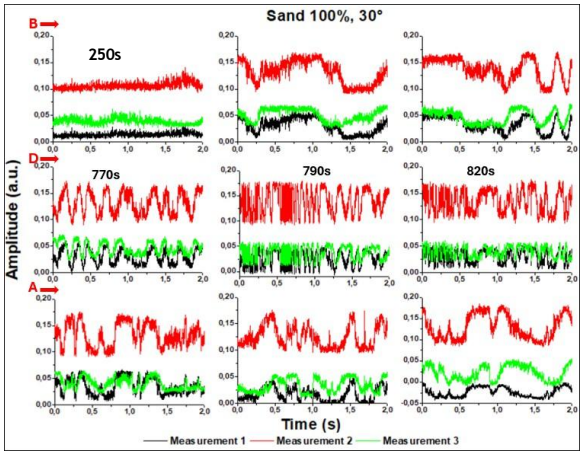


Figure 4: Time series signatures of Sand 100%, inclined at 30°. Moving from Top to Bottom of the column: Top row (B)- Before failure, Middle row (D)- During failure, and Bottom row (A)- After failure for a couple of failure events. Measurement 1, Measurement 2, and Measurement 3 are Stokes Parameters simultaneously recorded from 3 orthogonal axes.

Experiment 2 in Fig. 3 was conducted at 15° with a composition of 100% medium-grained sand. Peaks of amplitude variations shown in the top row (B: before) have a different pattern from those in the middle (D: during) and bottom row (A: after). The slope reached saturation just after 250 seconds, and surface erosion of the fine sediment was observed. The differences in amplitude variations result from the effect of oversaturation induced by rainfall on the gentle slope, which has less drainage over time. Compared to experiment 1, the erosion of fine sediments was slower in experiment 2, due to the difference in the angle of inclination. Once the soil reached saturation, a couple of cracks were formed at the bottom slope. The cracks connected to each other and moved vertically upwards into the middle slope; this also meant an increase in the length of the cracks. The first crack that formed at the bottom slope was 10cm long. An increase in saturation led to an extension of the crack to 23 cm, resulting in fractures with no mass movement.

The strain-induced events at 525 seconds and 640 seconds resulted in numerous fractures forming at the bottom and middle slopes. After the failure events, cracks that had formed before the failure continued to erode finer sediments as water infiltrated and weakened the soil. The system lost a significant amount of soil from the lower and middle parts of the slope due to erosion. The detachment of the top slope was observed, although it did not ultimately fail, as external forces, such as gravity, were not strong enough to result in a significant slope failure event. Little soil remained in the middle, and the bottom slopes were due to erosion before the slope reached equilibrium.

Fig.4 shows the same composition of Sand 100, inclined at a gentle angle of 30°. As the slope reaches saturation, the slope follows similar events observed in experiments 1 and 2 (Fig. 3). Notably, the saturation of 30° is delayed due to the infiltration versus runoff dynamics. A 30° slope is considered moderate, implying a balance between infiltration and runoff that creates a delay in saturation. The slope is steep enough to reduce infiltration efficiency but not steep enough to have cracks that rapidly channel water downslope. The first crack was observed in the 770s, when the slope began to fail, continuing through the 830s. Major erosion events were observed throughout the experiment. Compared to 45°, which lost half of the slope due to extensive erosion at 795s, 30° had prolonged instability.

The medium-grained sand used in these experiments was dry before the tap opened, meaning pore spaces contained air. As rainfall infiltration continued, the sand transitioned from an unsaturated frictional material to a fully saturated, fluid-like mass. The air in the pore spaces was displaced, and matric suction decreased. A decrease in matric suction leads to a decreased cohesion, which helps resist deformation. When the soil reached saturation, shear strength reduced, and the fibre data showed increasing vibration magnitudes and low-frequency signals, which reflected the onset of micro-failure. This is a point where sand behaves like a viscous material. These periodic processes occur during saturation, when the soil becomes oversaturated or reaches full saturation, causing pore water pressure between the soil particles to increase, thereby reducing frictional resistance. For steep slopes, this means rapid failures and mass flowing behavior is observed, e.g. at 630 seconds and 690 seconds in experiment 1 (Fig. 3). For the gentle slope of 15° and 30°, failure was slower and of moderate magnitude, occurring at 770 seconds with a peak-to-peak amplitude approximately 50% less than the magnitude on the steep 45° angle. The failure and slide processes indicate progressive rather than rapid deformation in gentle slopes.

The intensity spectrum curve in Fig. 5 shows the range before failure (b), during failure (d), and after failure (a). The frequencies detected before failure have lower intensity and have fewer components and magnitude than those during failure (d). Corresponding to Fig. 3 above, there were several failures in experiment 1, each one with a different intensity. This is evident in the number of high frequencies with higher intensities during failure. In Fig. 5a, it can be observed that after the failure event, the system had less acoustic energy as the remaining material re-established equilibrium and stability, as evidenced by the very low intensities.

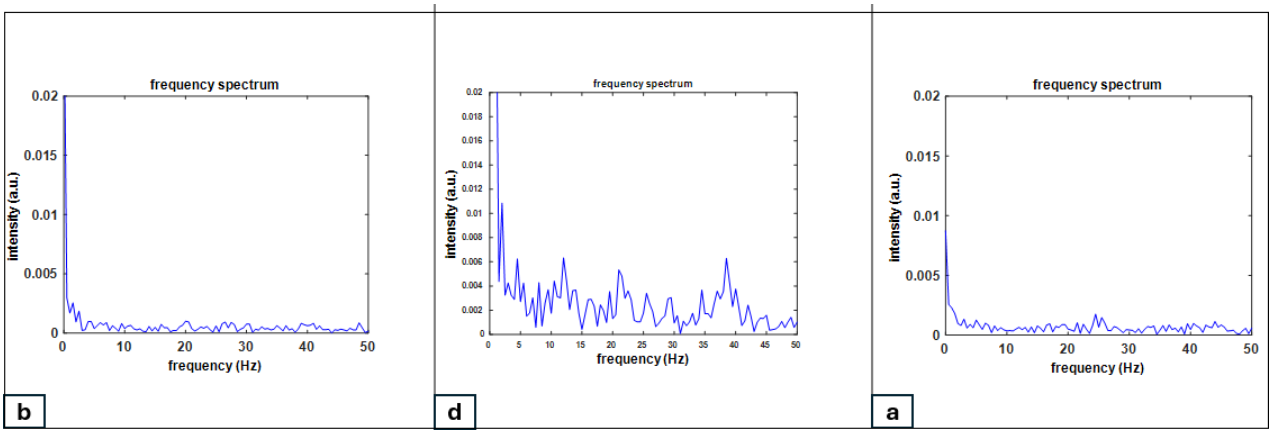


Figure 5: Spectral intensity distribution curves for Sand 100%, inclined at 45°. b- before failure, d-During failure, and a- After failure.

The frequencies detected before failure have lower intensity and have fewer components and magnitude than those during failure (d). Corresponding to Fig. 3 above, there were several failures in experiment 1, each one with a different intensity. This is evident in the number of high frequencies with higher intensities during failure. In Fig. 5a, it can be observed that after the failure event, the system had less acoustic energy as the remaining material re-established equilibrium and stability, as evidenced by the very low intensities.

The intensity spectrum curve in Fig. 6 and Fig. 7 shows the amount of strain applied to the fibre before (b), during (d), and after failure (a), in which the slope angle was 15° and 30°, respectively. The frequencies detected before failure had lower intensity than those during failure (d). The first sharp peak that occurred between 0-5 Hz, before failure, shifted to higher frequencies, detected from the inception of failure. As seen in Fig. 3 and Fig. 4, in the middle row, higher-frequency components with increased intensities were generated. Fig. 6(a) and 7(a) show energy dissipation after a failure event. This was evident in the lower intensities noticeable in the frequencies.

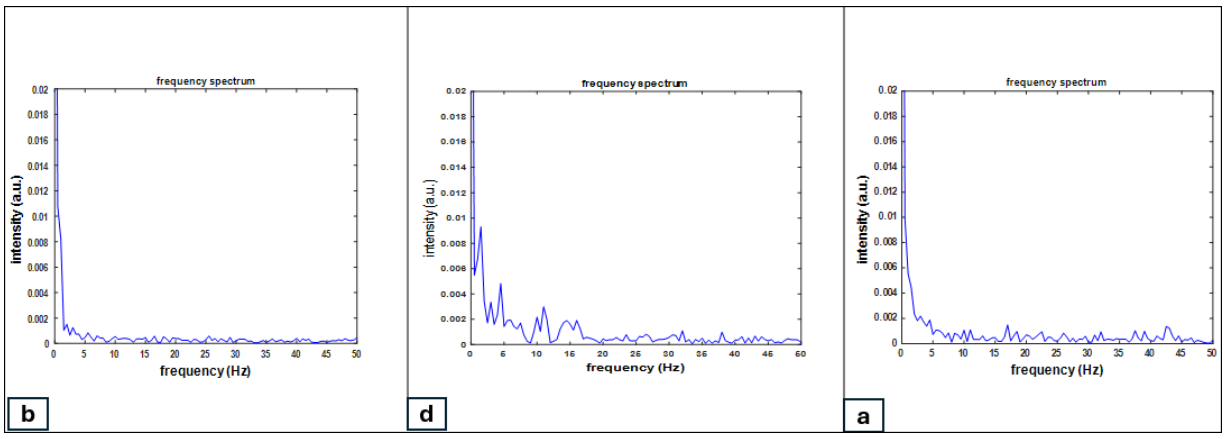


Figure 6: Spectral Intensity distribution curve for Sand 100%, inclined at 15°. b- before failure, d- during failure, a- after failure

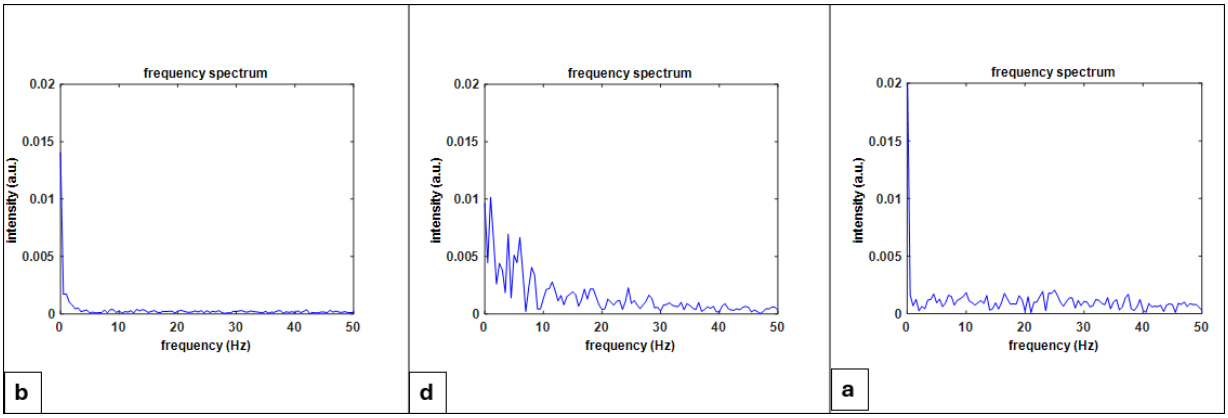


Figure 7: Spectral Intensity distribution curve for Sand 100%, inclined at 30°. b- before failure, d- during failure, a- after failure

Fig. 7(d) shows a couple of high peaks which represent the sequence of events that occurred in the 770s, 790s, and 820s. These repeated failure events result from the fact that 30° slope has a progressive infiltration and sufficient shear stress to repeatedly destabilize the slope. The shear stress of 45° is very high and it causes a catastrophic and early collapse of the slope. Furthermore, fig. 5b shows many peaks with high intensities; these are multiple simultaneous and sequential failure mechanisms that released large amounts of energy during the cracking, and extensive erosion from 525-960s.

In addition to the frequency and oscillations observed in the signals, there is a notable correlation between the peaks of amplitude variations in Figures 3 and 4, and intensity spectrum peaks in Figures 5, 6, and 7, respectively. In the results shown across all figures, we observe low-frequency vibrations detected before the slope failure event. This can be seen in Fig. 5b, 6b, and 7b, where the recorded signals exhibit extremely low intensities and slow oscillations before the failure, which gradually increase towards failure. The presence of these low-frequency vibrations and the monitoring of their changes indicate that the slope is gradually gaining momentum and instability before the primary failure event (Fig. 8). This observation suggests that the optical fibre sensor was able to detect the precursory signs of slope instability, providing an early indication of the impending failure.

The ambient signature of the slope, recorded for 2 minutes before the opening of the tap, is shown on Fig. 8b. The purpose of recording the ambient signature is to better distinguish between the disturbances caused by the slope or those caused by the outside influences. All experiments conducted showed similar ambient signatures. After the tap was opened, water flowed consistently, and the ambient signature changed to what is shown at 8d. Due to the influence of water, the signals show low-frequency and low-amplitude conditions. Fig. 8a is when the slope has reached saturation, and the dotted line represents the start of a mass-wasting event. The change in amplitude after the dotted line may be indicative of soil detachment from the slope. The scale difference for Fig. 8a is a result of a decrease in optical power because of elevated strain from high pore water pressure at saturation.

The ability to capture these low-frequency vibrations before the slope failure demonstrates the sensitivity and effectiveness of the optical fibre monitoring system. This information can be crucial for implementing timely mitigation measures and warning systems to prevent or reduce the impact of potential slope failures.

From these results, amplitude variations in time series signals, frequency distribution shifts in spectral plots and the correlation between time and frequency signals were computed on excel to support the findings of this study.

Time-domain signal analysis revealed that the mean polarization signal amplitude (*S*_mag) increased with slope inclination. Specifically, the mean S_mag values were 0.152 at 15°, 0.168 at 30°, and 0.179 at 45°, corresponding to a +10.5% increase from 15° to 30° and a +17.8% increase from 15° to 45°. Standard deviations, reflecting variability in strain response, were 0.036 at 15°, 0.045 at 30°, and 0.048 at 45°, indicating more erratic strain activity for steeper slopes.

The cross-correlation between time-domain amplitudes and spectral intensities was high across all slopes, with values of 0.87 for 15°, 0.91 for 30°, and 0.94 for 45°. This confirms that the amplitude of strain fluctuations closely corresponds to the frequency-domain energy distribution. This validates the effectiveness of the polarization-based optical fibre sensor used in this study in capturing both precursory and abrupt slope failure events.

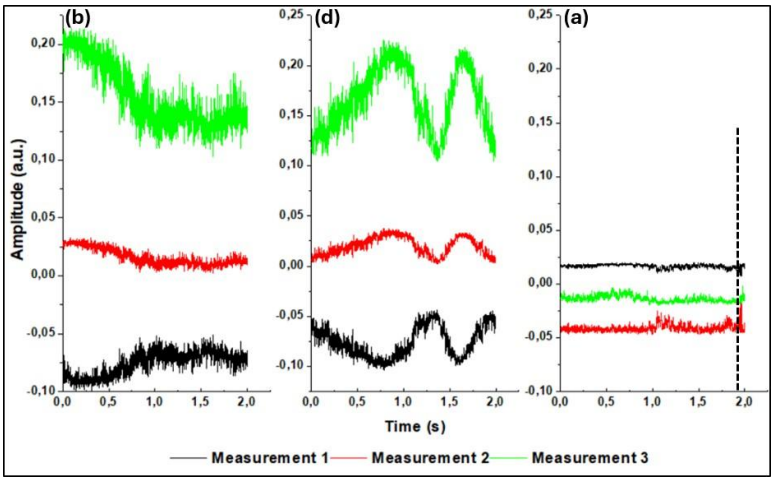


Figure 8: Signatures recorded before opening of the tap (b), during infiltration of the water (d), and after the slope has reached saturation (a).

V. Conclusion

This study successfully demonstrated that a polarization-based optical fibre strain sensor can be used as an early warning detection system for slope failure and landslide monitoring. The system reliably recorded precursory signals before failure, confirming its potential as an early warning detection mechanism for slope instability. Steeper slopes exhibited stronger, higher-frequency signals associated with rapid collapse, while gentler slopes generated lower-frequency, sustained vibrations reflecting progressive deformation. These patterns align with the geotechnical transition of sand from a rigid, frictional material in the unsaturated state to a viscous, fluid-like mass under full saturation, confirming the sensor's ability to capture both mechanical and hydrological drivers of slope failure. The use of polarization-based optical fibre offers continuous sensing over the entire fibre, with enhanced sensitivity to small strain variations.

Key findings include:

-Higher amplitudes and spectral intensities were captured for steeper slopes (30° and 45°), reflecting a more rapid failure mechanism.

-The sensor consistently recorded pre-failure events, which is a very critical feature for early warning systems.

Limitations of the study include the scale of the model, the use of homogeneous sand soils, and the resilience on artificial rainfall simulation. The complexity of natural slopes with heterogeneous soils, vegetation cover, and irregular rainfall events may not be fully replicated by the laboratory environment. Recommendations for future research include assessing this technique in the field and across diverse soil types and slope conditions. Development of automated algorithms for real-time failure detection and early warning detection. The reinforced learning and predictive modelling from AI-assisted detection can give us more reliable and accurate predictions for early warning.

Acknowledgment

We gratefully acknowledge the generous funding and support from the Mining Qualifications Authority (MQA), CSIR, and TELKOM Centre of Excellence.

References

[1] A. Gudmundsson. Displacement and strain. In: Rock Fractures in Geological Processes. Cambridge University Press; 2011:63-88.

[2] B. Finlayson, and I. Statham. Hillslope Analysis: Sources and Methods in Geography. Butterworths Publishers, 1980. ISBN 0 408 10622 0.

[3] M.A. Rahman et al. A review of Distributed Acoustic Sensing Applications for railroad condition monitoring. Mechanical Systems and Signal Processing. Elsevier. 2024.

[4] M.J.B. Alam, L.S. Manzano, R . Debnath, A.A. Ahmed. Monitoring slope movement and soil hydrologic behavior using IoT and AI technologies: A systematic review. Hydrology. 2024. 11, 111.

[5] K. Anjana, M. Herath, and J. Epaarachchi. Optical fibre sensors for geohazard monitoring- A review. *Measurement*, 235, 114846, 2024.

[6] P. Clarkson et al. . "Verification of a distributed fibre optic sensing slope stability monitoring solution." In *ARMA US rock mechanics/geomechanics symposium*, pp. ARMA-2022. ARMA, 2022.

[7] J.W. Kang et al. Automatic monitoring of seismogenic slope failure activity at Brienz (Switzerland) using distributed acoustic sensing and semi-supervised learning. 2024.

[8] D. Hill. "Distributed Acoustic Sensing (DAS): Theory and Applications," in Frontiers in Optics, OSA Technical Digest (online) (Optica Publishing Group). Paper FTh4E.1. 2015.

[9] J. Wang, S. Zhang, T. Lv, Y. Lui, K. Ji. Method application of distributed optical fibre Brillouin sensing in slope monitoring. 2024.

[10] M.J. Auflic et al. Climate change increases the number of landslides at the juncture of the Alpine, Pannonian, and Mediterranean regions. Scientific Reports, 2023. 13: 23085.

[11] M.J. Auflic, et. al. Landslide monitoring techniques in the geological surveys of Europe. Landslides, 2023, 951- 965.

[12] G. Rizzeli, M. Barla, and R. Gaudino. Algorithm Optimization for Rockfall Alarm System Based on Fibre Polarization Sensing. IEEE Photonics Journal, vol 15, 2023, pp 1-9.

[13] V. Ivanov et al. "Applicability of an Interferometric Optical Fibre Sensor for Shallow Landslide Monitoring-Experimental Tests." *Engineering Geology*. Vol 288. 2021

[14] A. Minardo, et al. Distributed Optical Fibre Sensor Applications in Geotechnical Monitoring. *Sensors 2021, Vol. 21, Page 7514*, *21*(22), 7514.

[15] A. Minardo et al. Long-Term Monitoring of a Tunnel in a Landslide-Prone Area by Distributed Optical Fibre Sensors. 2020. 6600-6603. doi: 10.1109/IGARSS39084.2020.9324181

OFDR fibre optic sensor for sinkhole sensing in a simulated environment

Francois J. Swanepoel
Department of Geoscience
Nelson Mandela Universtiy
Gqeberha, South Africa
fjswanepoel1998@gmail.com

James F. Jena
Department of Physics
Nelson Mandela University
Gqeberha, South Africa
s220399913@mandela.ac.za

Gaathier Mahed
Department of Geoscience
Nelson Mandela Universtiy
Gqeberha, South Africa
gaathier.mahed@gmail.com

David Waswa
Department of Physics
Nelson Mandela University
Gqeberha, South Africa
david.waswa@mandela.ac.za

Victor Chike Agbakoba
Department of Physics
Nelson Mandela University
Gqeberha, South Africa
Agbakobachike@gmail.com

***Abstract*--- A distributed OFDR (Optical Frequency Domain Reflectometer) sensor was designed for high-spatial-resolution monitoring of geological hazards. A TLS (Tunable Laser Source) with a sweep rate and range of ~1 GHz/s and 9.277 GHz, respectively, was used for this investigation. Using a 50 kHz sampling rate, distributed strain was derived through a DFT (Discrete Fourier Transform) with window sizes of 20,000,000 and 256. To test and calibrate the system, a 1 kg mass was positioned at both the beginning and end of the sensing fibre, with results showing localized strain at 0.08 m and 3.85 m along the FUT (Fibre Under Test). Building on this, a laboratory-scale simulated sinkhole environment was developed, incorporating both fibre optic and capacitance-based soil moisture sensors. Results demonstrated a sharp reduction in soil moisture content after peak saturation, coinciding with outflow and the onset of collapse. Distributed strain signatures revealed a characteristic "saddle" pattern, consistent with tensile and compressive strain zones during sinkhole initiation, progression, and failure. These findings validate the experimental setup and highlight the potential of OFDR sensing for detecting precursory signals of geohazard formation. Future work will expand on post-processing techniques and multi-sensor integration to enhance quantitative measurements and field applicability**

Keywords--- OFDR, Geological Disasters, Distributed sensing, Remote monitoring, Environmental management.

I. INTRODUCTION

A global landslide database compiled from literature sources records 410 landslide dam failures occurring between 1900 and 2020 [1]. Within this dataset, rainfall is identified as the leading trigger (42%–44% of cases), followed by seismic activity (20%–29% of cases) [1]. On a broader scale, approximately 75 billion tons of soil are displaced worldwide each year, with Africa, Asia, and South America accounting for the majority of this figure [2]. Due to the environmental and social impacts of such events, landslides remain one of the most extensively studied subjects in geology, geomorphology, and environmental management [3]. To address and reduce these impacts, there has been a notable shift away from conventional monitoring techniques and equipment [4]. Current approaches largely rely on radar, high-resolution cameras, and geophones [5]. However, these methods are limited in their ability to capture the small-scale processes that precede major geological disturbances [4–7]. The rapid advancement of optoelectronics and fibre-optic communication technologies has led to the development of a new class of sensors [8]. These optical sensors operate on the principle of optical reflectometry [9], where optical signals are used to measure strain induced by external acoustic interactions along a sensing fibre [8], [9]. This technology is particularly attractive because it enables high-resolution, distributed sensing—a capability previously unavailable in geological monitoring and modelling [4–7], [9–12]. Distributed Acoustic Sensing (DAS) not only provides coverage across large areas but is also compact, resistant to electromagnetic interference, and functional under hazardous conditions [4–10], [12], [13]. Remarkably, a single DAS interrogator can replace hundreds of conventional geophones, offering resolutions that range from millimetres to metres depending on the sensing configuration [4–10], [12], [14], [15]. For this investigation, an Optical Frequency Domain Reflectometer (OFDR) sensor was developed with the specific aim of monitoring and characterizing the processes that lead to the formation of sinkholes.

II. METHODS AND DESIGN

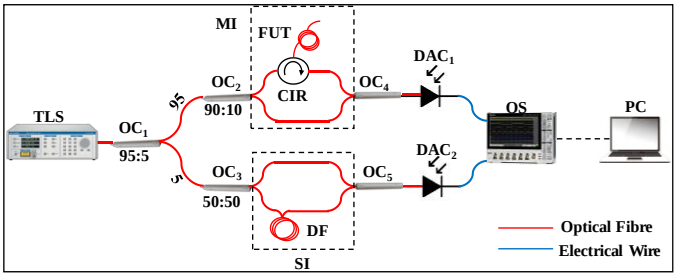


Fig. 1. Schematic diagram of the OFDR set up used.

A. *OFDR Design*

The experimental set up for the OFDR can be seen in Fig 1. The OFDR is composed of a 2-interferometer system, named MI (Main Interferometer) and SI (Secondary Interferometer). A TLS (Tunable Laser Source) is used to produce a wavelength – swept optical signal centered around 1550 nm, with a sawtooth pattern. The output of the TLS was split using a 95/5 optical coupler (OC_1), with a 95 output to the MI. High intensity signals in MI are further divided using a 90:10 OC_2. Backscattered signals from the circulator (CIR) are recombined with the forward scattered signals in the MI at OC_4. Similarly, the SI splits optical signals at OC_3, with forward scattered signals recombining at OC_5. Two separate Data Acquisition Cards (DAC_1 and DAC_2) are used to convert optical signals into electrical signals, which are then displayed on an Oscilloscope (OS). The SI is used as a compensation method to mitigate the effect of nonlinear frequency sweep that the MI may experience. Finally, to compensate for the two–way travel time of the backscattered signal in the FUT (Fibre Under Test), the DF (Delay Fibre) was kept at twice the length (81.8 m) of the FUT (40.9 m). The FUT is made up of an approximate DF length of 37 m combined with a sensing region of 3.9 m.

B. *Non – Linear Sweep correction*

An inherent non–linear sweep is produced by the TLS [14 - 16]. Non–linear sweeps cause the energy of the Rayleigh-backscattered light to spread [15]. As the reflection peaks widen, the overall spatial resolution of the system decreases [15]. To correct for this, the beating signal from the SI is used to resample the raw measured and reference signals from the MI. The electric field intensity of the generated optical signals by the TLS may be expressed as [14]:

$$I(t) = I_o \cos[2\pi \tau_z \upsilon(t)] \quad (1)$$

where I_o and $\upsilon(t)$ are the amplitude and the instantaneous frequency of the beat signals in the SI (Equation 1).

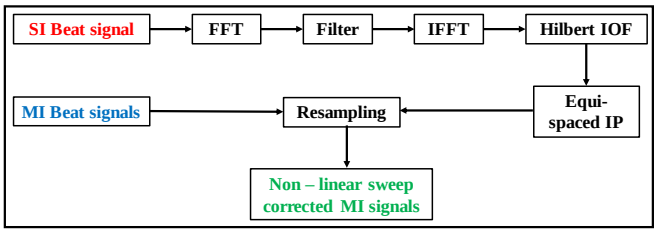


Fig. 2. Workflow used for non – linear sweep correction

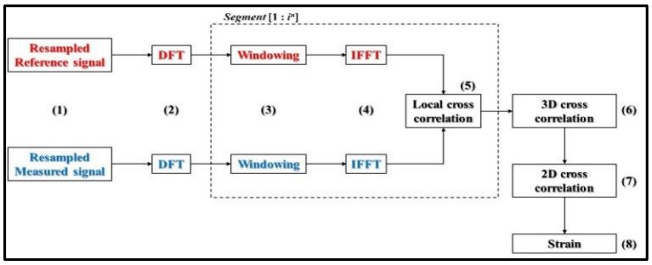


Fig. 3. Demodulation procedure used to calculate Rayleigh Backscattered Shifts (RBS).

Raw SI signals are first converted to the frequency domain using an FFT (Fast Fourier Transform). A digital filter in the frequency domain is then applied to the axillary peak, while the remainder of the signal is zeroed [16]. Thereafter, the filtered signal is converted back into the time domain using an IFFT (Inverse Fast Fourier Transform). The Hilbert transform is then applied to the time domain filtered signal for calculation of the IP (Instantaneous Phase) of the SI (Equation 2) [14], [16].

$$\Phi(t) = 2\pi \tau_z \upsilon(t) = \arctan\left(\frac{H[I(t)]}{I(t)}\right) \quad (2)$$

Further expanding of $\Phi(t)$ leads to the calculation of the instantaneous optical frequency (IOF) (Equation 3) [14], [16].

$$\upsilon(t) = \left(\frac{1}{2\pi \tau z}\right) \arctan\left(\frac{H[I(t)]}{I(t)}\right) \quad (3)$$

Finally, the IP was rearranged in equally spaced intervals which were used to resample the raw time domain signals using spline interpolation (Fig 2) [16].

C. *Cross correlation processing*

A detailed workflow of the signal processing methods used throughout the investigation is described in Fig 3. Before experimentation, a reference signal without any strain applied was measured, followed by the measurement of a signal with apparent strain applied across the sensing region. After resampling both the reference and measured signals, signals could be converted to the spatial domain using a DFT (Discrete Fourier Transform). The auxiliary peak in the frequency domain is recognized as the frequency representing the end of the sensing region. Realizing this, the conversion to the spatial domain based on the length of the FUT could be made. Windows are used to create segments, each representing a small measurable segment along the sensing region. Theoretical spatial resolution (Δz) of the device is indirectly related to the sweep range (Δv) of the OFDR and is given by:

$$\Delta z = \frac{c}{2n\Delta v} \quad (4)$$

Spatial resolution (ΔL) of the OFDR is thus a product of the chosen window size (N) (Equation 5).

$$\Delta L = N\Delta z \quad (5)$$

Extraction of two windows from the same spatial position are converted into the time domain via an IFFT and cross correlated to find the positional lag. These values represent the time value with which the reference and measured signals differ at a particular spatial index and are caused by applied strain. Windowing cross correlation between the reference and measured signal at every spatial position was then stacked to produce a 3D cross correlation image [14], [15], [17]. A time lag peak detection algorithm is used to extract the 2D lag at every sensing point along the FUT. Since lags represent the difference in two-way travel time at each respective location

along the FUT, time lag (Δt) can be converted to a spatial lag (ΔL) using the following relationship:

$$\Delta L = \Delta t \left(\frac{c}{n}\right) \quad (6)$$

Where n is the refractive index of the single mode fibre (n =1.5) and c the speed of light in a vacuum. Since strain (ε) is defined as the ratio between the relative change in length (ΔL) over the initial change in length (L), the relative strain (ε) at any given point along the FUT can be derived as follow:

$$\varepsilon = \frac{\Delta t\, c}{n\, L} \quad (7)$$

D. Simulated sinkhole environment:

Experiments conducted were done in a simulated sinkhole model with dimensions of 100 cm × 50 cm × 30 cm (Fig 4). The model contained inlet and outlet flow pipes which facilitated the movement and deposition of moisture and sediment, later creating a sinkhole. A fibre connector was also installed to allow for seamless integration of the fibre optic sensor inside the simulated environment.

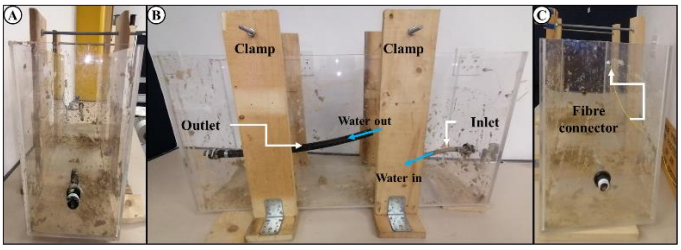


Fig. 4. Simulated sinkhole environmental model. A) Back view of the model. B) Side view of the model. C) Front view of the model.

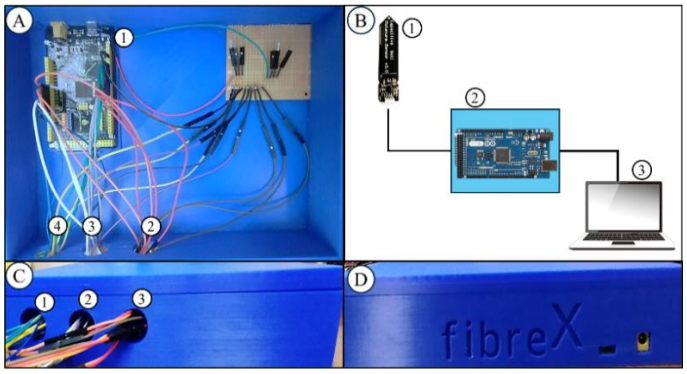


Fig. 5. A) Top view of the electronic housing containing the Arduino mega board (1) and wiring for the moisture sensors (2-4). B) Schematic of the electronic sensor set up. (1) Capaciatance – based soil moisture sensor connected to an Arduino Mega board (2) enclosed in a housing and connected to a PC (3). C) Front view of the electrical housing with wiring for moisture sensors (1-3). D) Back view of the electical housing.

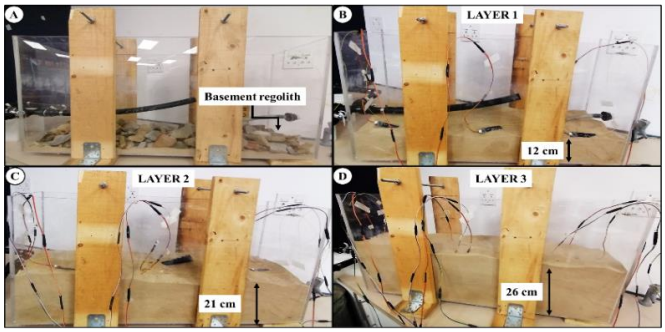


Fig. 6. Material layering with basement regolith (A), layer 1 (B) at 12 cm from the bottom, layer 2 (C) at 21 cm from the bottom and layer 3 (D) at 26 cm measured from the bottom.

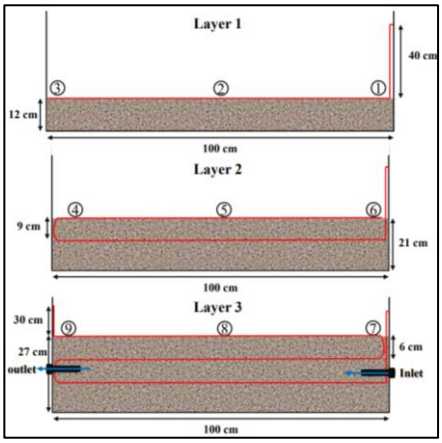


Fig. 7. Sensor placement of fibre optic sensor (in red) and soil moisture sensors (numbered).

E. Electronic sensors:

In addition to the fibre optic sensor, 9 inexpensive capacitance – based soil moisture sensors were used to validate the moisture content throughout the experiment. Electronic sensors ran from an Arduino Mega board which was connected to a PC (Personal Computer) (Fig 5).

F. Experimental procedure:

In preparation for the experiment, a few preliminary steps were taken. These included proper material layering, sensor placement and sensor calibration.

1) Material layering:

Before the unconsolidated material could be placed into the model, a layer of consolidated material simulating a basement regolith, was placed at the bottom of the model (Fig 6A). Thereafter, three layers of unconsolidated sandy material were placed, at heights 12 cm, 21 cm and 26 cm (Fig 6B-D).

2) Sensor placement:

On every layer, 3 soil moisture sensors were placed equidistantly. Additionally, the fibre optic sensor was placed in an "S" like pattern curving upward, starting from layer 1 (Fig 7).

3) Sensor calibration:

After the burial of the fibre sensor inside the model, a reference measurement was taken. Reference measurements would reflect fibre sensor under dry, stable conditions.

Moisture sensors were calibrated by adding sediment to a crucible and saturating it with water. Thus, moisture sensors could be calibrated based on the raw value sensor output under saturated conditions.

III. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. *Benchmark results:*

As a proof of concept, a 1 kg mass was placed at the start and end of the FUT. To validate the design and procedures undertaken, the experiment was aimed at confirming the strain at these locations. After interpolation of the IP of the SI, the IOF was further calculated using Equation 3. The sweep rate and sweep range were determined for the system using this method (Fig 8). Best results were achieved using a DFT and a window size of 20,000,000 and 256 respectively. Furthermore, overlaps of 75% between successive window sizes were used for strain estimations. The sweep rate and sweep range for the system was found to be ~ 1 GHz/s and 9.277 GHz. After resampling, a clear peak shift between the two signals in the frequency domain was observed as indicated in Fig 9. After cross correlation between the signals in the spatial domain, a 2D RBS was achieved for both weight–applied instances (Fig 10). Clear shifts may be seen relating to the positions of strain on the FUT. Strain values for both locations measured at -11581.7 με and -10621.8 με for the start and end of the fibre respectively. Locations of these disturbances correspond to the placement of the 1 kg mass, with localized strain at the start (0.08 m) and at the end (3.85 m) of the FUT.

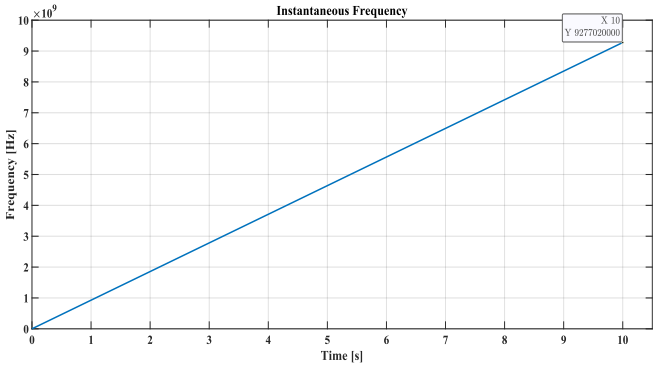


Fig. 8. Interpolated IOF of the AI

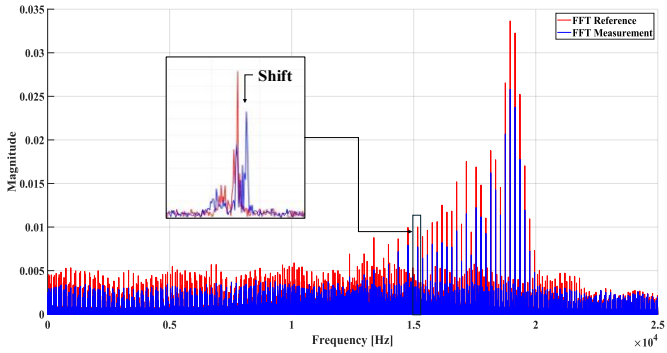


Fig. 9. Resampled raw signals in the frequency domain. The data displays a peak shift.

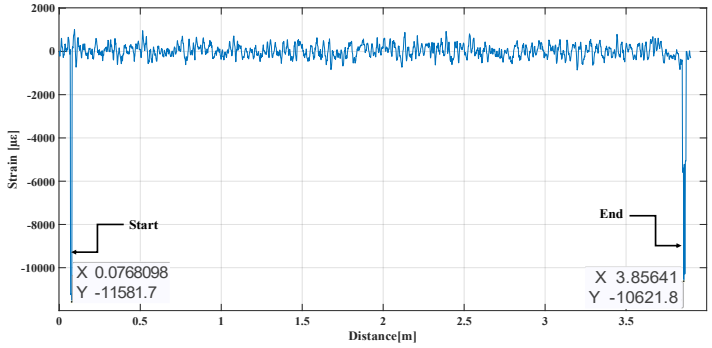


Fig. 10. RBS along the FUT. Clear compressive strain peaks observed at the start and end of the FUT

TABLE I. GRAIN SIZE ANALYSIS RESULTS OF SIEVED SAMPLE

Grain size [mm]	Weight per aperture size [g]	Cumulative Weight [g]	Finer [%]
< 0,065	0	0	0
0,065	0,009	0,009	0,0009
0,075	0,05	0,059	0,0059
0,09	0,95	1,009	0,1
0,125	37,86	38,869	3,89
0.180	224,03	262,899	26,3
0,25	676,69	939,589	93,9
0,355	54,82	994,409	99,5
0,5	5,04	999,449	99,98
1	0,17	999,619	100
Error [%]		0,038088	

TABLE II. CHARACTERISTICS OF UNCONSOLIDATED MATERIAL

Grain Diameter [mm]	D10 [mm]	D30 [mm]	D60 [mm]	Uniformity Coefficient [Cu]	Curvature Coefficient [Cc]	Hydraulic conductivity [cm/s]	Error [%]
0,186	0,14	0,184	0,215	1,535	1,125	2,39e-2	3,81e-2

B. *Sediment classification:*

A simple sieve test was conducted to establish the characteristics and physical properties of the unconsolidated material used for the experiment. Sieve tests confirmed the material to be primarily composed of sand grains with sizes ranging from 0,355 mm – 0,125 mm (Table 1). Furthermore, effective particle size fractions, D10, D30 and D60 as well as the Uniformity Coefficient (CU) and the Coefficient of Curvature (CC) were also calculated for this investigation (Table 2). Hydraulic conductivity was calculated using the software, SizePerm. Since CU < 6 and CC lie within the range of 1 < CC < 3, the material could be classified as a medium – fine grained, well sorted, uniformly graded sediment (Table 2). The lack of very fine-grained material, such as clay, allowed for the further classification of the induced sinkhole as a suffusion sinkhole, since material is removed at depth through the perforated outlet pipe.

C. *Electronic sensors:*

On average, moisture sensor data shows an increasing trend, as water was injected into the system, reaching an average soil water content of 64.5% (Table 3)(Fig 11). Water started pouring from the system during the 12-minute mark, corresponding to the followed decrease in soil water content (60.8%). This time also coincided with a significant observable collapse event.

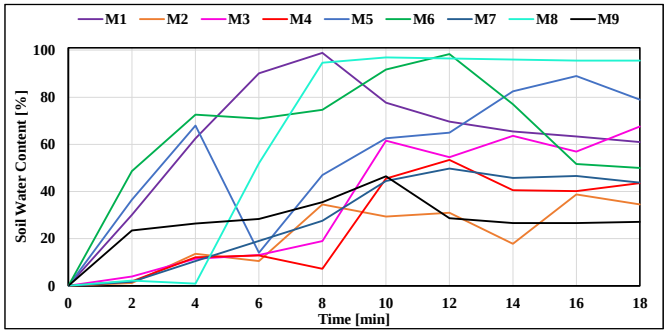


Fig. 11. Soil water content (%) response every 2 minutes for moisture sensors 1 to 9 (denoted as M1 to M9)

TABLE III. INDIVIDUAL MOISTURE SENSOR RESPONSE AND GLOBAL AVERAGE

Time [s]	M1 [%]	M2 [%]	M3 [%]	M4 [%]	M5 [%]	M6 [%]	M7 [%]	M8 [%]	M9 [%]	Average [%]
0	0	0	0	0	0	0	0	0	0	0
2	30,1	1,2	4,0	1,9	36,5	48,6	1,7	2,2	23,5	16,6
4	62,5	13,6	11,5	12,1	68,0	72,6	10,5	0,9	26,4	30,9
6	90,2	10,5	13,0	12,9	14,0	70,9	19,0	52,0	28,3	34,5
8	98,8	34,5	19,0	7,2	47,0	74,7	27,6	94,7	35,5	48,8
10	77,7	29,4	61,6	45,5	62,6	91,8	44,5	96,9	46,5	64,5
12	69,6	31,0	54,5	53,4	65,0	98,3	49,7	96,4	28,7	60,8
14	65,5	17,8	63,6	40,5	82,5	77,0	45,7	96,0	26,6	57,3
16	63,4	38,8	56,9	40,2	89,0	51,7	46,6	95,6	26,6	56,5
18	61,0	34,5	67,6	43,6	79,0	50,0	43,8	95,6	27,1	55,8

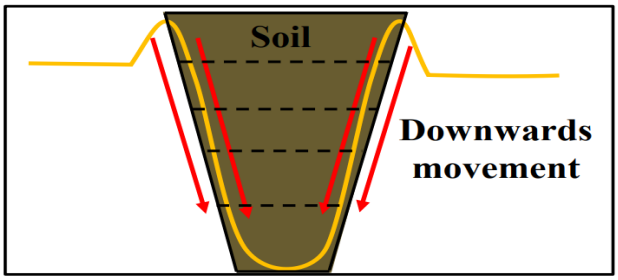


Fig. 12. Pictorial representation of the "saddle" strain shape occuring during sinkhole creation and progression

D. Collapse signitures:

Strain patterns associated with sinkholes exhibit a characteristic "saddle" shaped pattern (Fig 12) [18]. This characteristic shape is caused by vertical shear band occurring on both sides of the sinkhole, facilitating the formation of tensile strain peaks along the edges of collapse strain zones [19]. As sinkholes collapse inward, pull – back tension is applied on the fibre sensor at the soil – fibre interface [20]. Furthermore, compressive strain is applied in the center as overburdened soil collapse inwards, thus leading to the development of the "saddle" shapes [21].

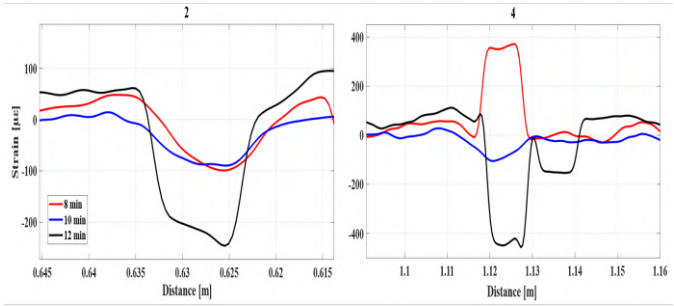


Fig. 13. Distributed strain sinkhole signatures at locations 2 and 4 according to Fig 7, for time 8 minutes (Red), 10 (Blue), and 12 (Black)

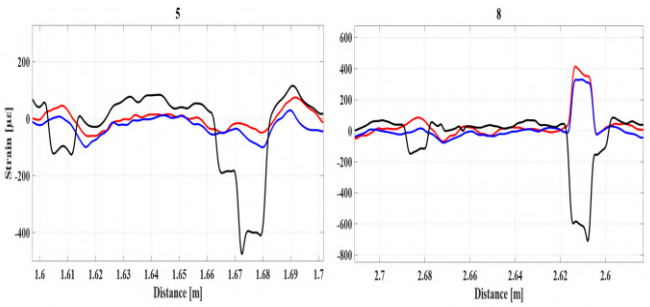


Fig. 14. Distributed strain sinkhole signatures at locations 5 and 8 according to Fig 7, for time 8 minutes (Red), 10 (Blue), and 12 (Black)

Investigating the time of the significant change in moisture content relating to an observable collapse event, corresponding notable distributed strain signatures were plotted for further analysis (Fig 13 and Fig 14). DFT and window sizes similar to what was used during the preliminary tests were used for analysis of data. Features resembling saddle–like shapes, reminiscent of cavity formation and sinkhole collapse, were observed at 12 minutes. However, these features were notably absent for the distributed strain measured at 8 and 10 minutes. At the 8-minute mark, increased tensile strain was observed across all monitoring locations in Fig 13 and 14. As material is washed away, a cavity starts forming. Cavity formation reflects the progressive buildup of vertical shear stresses such as those observed at 8 minutes [19]. After which reduced strain recorded at 10 minutes corresponded with a period of creep compaction of cover material [21]. Over time, as more material is washed away at depth, vertical shear bands extend to the surface as cover material collapse inward [19]. The eventual collapse at 12 minutes coincided with the double-peak trough shape reminiscent of sinkhole collapse.

IV. CONCLUSION

This study details the development and successful application of an experimental OFDR sensor for monitoring sinkhole-forming processes in a controlled laboratory environment. The proposed signal processing workflow, which utilizes a DFT and a window size of 20,000,000 and 256 respectively, enabled the successful extraction of benchmark tests and sinkhole collapse signals. Future work might include the optimization of window sizes and frequency domain filtering methods for further improvement of the dynamic range and signal to noise ratio of the sensing apparatus.

The results validated both the proposed workflow and the OFDR sensor design. Compressive strain peaks at the locations of the placed weights along the FUT. Electronic sensor data showed a clear decrease in average soil water content at the 12-minute mark, which correlated with the observed collapse.

Collapse strain signatures, identified by their characteristic "saddle-like" shape, were observed during this same period. These shapes are attributed to the development of vertical shear strain, similar to that observed at 8 minutes. These signatures evolve as creep compaction occurs, with weak vertical shear bands extending toward the surface and greater compressive strain localized in the signature troughs. The final stage of collapse, where overhead material moves inward, is a result of

this combined creep compaction and vertical shear band extension. These changes directly corresponded to the distributed strain calculated at the 10- and 12-minute marks.

Future work should explore different types of unconsolidated materials and incorporate more precise soil moisture sensors. The inclusion of other sensors, such as temperature or particle image velocimetry, is also recommended to provide a more comprehensive analysis.

ACKNOWLEDGEMENT

The authors would like to gratefully acknowledge; the Telkom Centre of Excellence, CSIR and HIRAX for providing financial support for this research.

REFERENCES

[1] X. Fan, *et al*., "The formation and impact of landslide dams – State of the art," *Earth-Science Reviews*, vol. 203. Elsevier B.V., Apr. 01, 2020. doi: 10.1016/j.earscirev.2020.103116.

[2] J. Ananda and G. Herath, "Soil erosion in developing countries: A socio-economic appraisal," *J Environ Manage*, vol. 68, no. 4, pp. 343–353, Aug. 2003, doi: 10.1016/S0301-4797(03)00082-3.

[3] I. Bollati, M. Della Seta, M. Pelfini, M. Del Monte, P. Fredi, and E. Lupia Palmieri, "Dendrochronological and geomorphological investigations to assess water erosion and mass wasting processes in the Apennines of Southern Tuscany (Italy)," *Catena (Amst)*, vol. 90, pp. 1–17, Mar. 2012, doi: 10.1016/j.catena.2011.11.005.

[4] T. de Wit and G. Olivier, "Imaging and monitoring tailings dam walls with ambient seismic noise," in *Proceedings of the 21st International Seminar on Paste and Thickened Tailings*, Australian Centre for Geomechanics, Perth, 2018, pp. 455–464. doi: 10.36487/acg_rep/1805_37_de_wit.

[5] D. Inaudi and E. Régis Blin, "Monitoring dams with distributed fiber optic sensing," 2016.

[6] B. G. Gorshkov, *et al*., "Scientific Applications of Distributed Acoustic Sensing: State-of-the-Art Review and Perspective," *Sensors*, vol. 22, no. 3. MDPI, Feb. 01, 2022. doi: 10.3390/s22031033.

[7] T. Parker, S. Shatalin, and M. Farhadiroushan, "Distributed Acoustic Sensing-a new tool for seismic applications," 2014. [Online]. Available: www.firstbreak.org

[8] S. Yin, P. Ruffin, and F. Yu, *Fiber Optic Sensors*, 2nd Edition. CRC Press, 2008.

[9] Y. Li, M. Karrenbach, and J. Ajo - Franklin, *Distributed Acoustic Sensing in Geophysics: Methods and Applications*, 1st Edition. Wiley, 2022.

[10] Z. He and Q. Liu, "Optical Fiber Distributed Acoustic Sensors: A Review," *Journal of Lightwave Technology*, vol. 39, no. 12. Institute of Electrical and Electronics Engineers Inc., pp. 3671–3686, Jun. 15, 2021. doi: 10.1109/JLT.2021.3059771.

[11] T. Parker, S. Shatalin, and M. Farhadiroushan, "Distributed Acoustic Sensing-a new tool for seismic applications," 2014. [Online]. Available: www.firstbreak.org

[12] X. Bao and L. Chen, "Recent Progress in Distributed Fiber Optic Sensors," *Sensors (Switzerland)*, vol. 12, no. 7, pp. 8601–8639, Jul. 2012, doi: 10.3390/s120708601.

[13] Z. Wang, B. Lu, Q. Ye, and H. Cai, "Recent progress in distributed fiber acoustic sensing with Φ-otdr," *Sensors (Switzerland)*, vol. 20, no. 22. MDPI AG, pp. 1–26, Nov. 02, 2020. doi: 10.3390/s20226594.

[14] Q. Wang, *et al*., "Improving OFDR Distributed Fiber Sensing by Fibers With Enhanced Rayleigh Backscattering and Image Processing," *IEEE Sens J*, vol. 22, no. 19, pp. 18471–18478, Oct. 2022, doi: 10.1109/JSEN.2022.3197730.

[15] C. Liang, Q. Bai, M. Yan, Y. Wang, H. Zhang, and B. Jin, "A Comprehensive Study of Optical Frequency Domain Reflectometry," *IEEE Access*, vol. 9. Institute of Electrical and Electronics Engineers Inc., pp. 41647–41668, 2021. doi: 10.1109/ACCESS.2021.3061250.

[16] M. Badar, P. Lu, M. Buric, and P. Ohodnicki, "Integrated auxiliary interferometer for self-correction of nonlinear tuning in optical frequency domain reflectometry," *Journal of Lightwave Technology*, vol. 38, no. 21, pp. 6097–6103, Nov. 2020, doi: 10.1109/JLT.2020.3007703.

[17] P. Li, *et al*., "High-Spatial-Resolution Strain Sensor Based on Distance Compensation and Image Wavelet Denoising Method in OFDR," *Journal of Lightwave Technology*, vol. 39, no. 19, pp. 6334–6339, Oct. 2021, doi: 10.1109/JLT.2021.3095851.

[18] F. Gutiérrez, *et al*., "The Application of Distributed Optical Fiber Sensors (BOTDA) to Sinkhole Monitoring. Review and the Case of a Damaging Sinkhole in the Ebro Valley Evaporite Karst (NE Spain)", *Engineering Geology*, 325, 2023. Available at: https://doi.org/10.1016/j.enggeo.2023.107289

[19] T. Möller, *et al*., "Distributed Fiber Optic Sensing for Sinkhole Early Warning: Experimental Study", *Geotechnique*, 73(8), pp. 701-715, 2022. Available at: https://doi.org/10.1016/j.enggeo.2023.107289

[20] Y.X. Goa, *et al*., "Performance Evaluation of Distributed Strain Sensing Nerves for Monitoring Ground Collapse: A laboratory Study", *Measurement: Journal of the International Measurement Confederation*, 217, 2023. Available at: https://doi.org/10.1016/j.measurement.2023.113100

[21] L. Cheng, *et al*., "A Distributed Fiber Optic Monitoring Method for Ground Subsidence Induced by Water Pipeline Leakage", *Optical Fiber Technology*, 81, 2023 Available at: https://doi.org/10.1016/j.yofte.2023.103495.

Francois Jakobus Swanepoel is a PhD student at the University of Cape Town, Cape Town, South Africa.

A Review of Maximum Power Point Tracking Methods for Wind Energy Conversion Systems

Floyd L. Mabiala1
Department of Computer Science
University of the Western Cape
Cape Town, South Africa
3755563@myuwc.ac.za

Clement N. Nyirenda
Department of Computer Science
University of the Western Cape
Cape Town, South Africa
cnyirenda@uwc.ac.za

Mohamed T.E. Kahn
Energy Institute
Cape Peninsula University of Technology
Cape Town, South Africa
khant@cput.ac.za

***Abstract*—With the growing adoption of wind energy within distributed generation systems and isolated microgrids, optimizing the performance of wind turbines becomes essential. Wind turbines must operate at their peak energy conversion efficiency to meet load requirements. However, their performance is influenced by environmental factors. To address this challenge, integrating Artificial Intelligence (AI) models into wind energy conversion systems (WECS) is crucial. These intelligent systems enable precise tracking of wind power, leading to improved overall efficiency and performance of wind turbines. To optimize the tracking of wind turbine power, several AI-based maximum power point tracking (MPPT) methods for WECS have been reported in several articles during the last few years. Most of them discussed conventional and simple intelligent methods, such as Artificial Neural Network (ANN) and fuzzy logic systems. There is, however, no comprehensive study in the literature about various Neuro-Fuzzy System (NFS) methods in WECS. Hence, this work presents a thorough review by presenting conventional MPPT techniques for WECS, culminating in the discussion of Neuro-Fuzzy Systems for MPPT in WECS. In this review, these NFS methods have been categorized into two groups, namely, single and hybrid MPPT methods. The advantages and disadvantages of each group have been presented. This work revealed a high potential for NFS-based MPPT methods because they offer a balance between learning capability, stability of the system responses, accuracy, and complexity, and therefore, better performance in the WECS. Thus, they can be seen as a promising research area to devise robust MPPT methods for wind energy systems. Finally, the future objective would be to reduce the computational burden while maintaining overall efficiency.**

***Index Terms*—AI, ANN, ANFIS, Fuzzy logic, Machine learning models, MPPT, Wind energy conversion system**

I. Introduction

The use of renewable energy sources is rapidly expanding to meet the rising global energy demand, aimed at overcoming the drawbacks of fossil fuels, such as their high cost, depletion, and greenhouse effect [1]. Renewable energy sources, such as wind energy, are gaining immense popularity, owing to their environmental friendliness and availability [2]. Wind energy is obtained from the turbine that converts the kinetic wind energy into mechanical energy, which is then converted into electrical power [3]. A remarkable increase in the installation of wind power systems is happening globally, and the overall installation is scheduled to take 20% of the global energy demand by 2030, the cumulative wind power capacity reached about 930GW at the end of the year 2023 and about 1500GW by the year 2030 [1]. However, the output power of the wind energy conversion system (WECS) is variable due to the fluctuating nature of wind speed [4]. The wind turbine operates in a specific range of wind speeds, leading to the system shutdowns for any other wind speed. This leaves a challenge in estimating the maximum power attained by the turbine. Given the complex nature of wind energy systems and their significant reliance on environmental conditions, it becomes essential to integrate control mechanisms. These systems aim to optimize the operation of the wind turbines (WTs) and maximize power generation from the wind turbines [5]. Such a control system is referred to as the maximum power point tracking (MPPT) controller, which works by adjusting the operating parameters of the system to match the MPP of the wind turbine [3]. There are two types of MPPT control methods available, known as conventional and intelligent methods [2] [6]. *(i) Conventional methods*, use complex mathematical equations and require information about the wind turbine to estimate the MPP of the wind power output. However, they suffer from power fluctuations and overshooting around the optimal point [7], slow response times when wind conditions change rapidly, model dependency, meaning relying on mathematical models of the wind turbine or wind characteristics [8] as well as a lack of adaptability [4] [9], which implies that they are generally static and do not adapt well to changing wind conditions. *(ii) Intelligent MPPT methods* use advanced artificial intelligence methods and soft computing reasoning methods to estimate the maximum power of the wind turbine. Therefore, intelligent methods are under investigation to address the mentioned challenges. Moreover, intelligent methods such as neuro-fuzzy systems (NFS) are emerging as particularly promising [10]. These AI systems merge the learning and generalization strengths of neural networks (NNs) with the rule-based reasoning power of fuzzy logic [11]. They can learn the complex relationships between the operational parameters of Wind turbines and the power output. Equipped with this knowledge, they can track in real-time the MPP. Hence, this fusion of capabilities makes them ideally suited for MPPT applications. In the literature, most reviews focus on conventional methods and simple intelligent methods, including fuzzy logic, and NNs [6] [11]. The author

in [11] reported on modified sensor-less methods such as P&O and INC methods. The reviewers in made an overview of conventional, intelligent optimization methods, as well as their hybrid methods [2]. It shows the importance of implementing variable, adaptive, and hybrid step sizes in estimating the power output. Navarrete et al [12] present a review of expert systems implemented in the pitch control of wind turbines. Belkhiri et al. [13] briefly reviewed AI methods in wind turbines and provided a thorough analysis of both Artificial Neural Networks (ANN) and Fuzzy Logic (FL) used in WECS. However, there is no comprehensive study in the literature about various neuro-fuzzy system methods in wind turbines. The current literature does not clarify how the accuracy of tuning the parameters and the complexity of the design can be overcome. Moreover, it does not emphasize the improvement made by neural fuzzy systems in determining the exact values of the membership function, and it also does not include other hybrid neuro-fuzzy systems [2]. Therefore, this study provides comprehensive information on the use of neural fuzzy systems and categorizes intelligent methods into two categories, namely, single and hybrid methods, and focuses on the advantages and disadvantages of each method.

The main contributions of the present survey can be summarized as follows: (1) A review of MPPT methods based on NFS for WECS. It begins by introducing an overview of the Wind energy system and the basic concepts of MPPT. (2) A detailed presentation of NFS methods applied for MPPT in WECS starts by discussing the most used conventional methods and then the different types of NFS that have been used for MPPT in WECS, including adaptive neuro-fuzzy inference systems (ANFIS), type-2 fuzzy neural networks (T2FNNs), optimized Neuro-Fuzzy Systems, and Wavelet neuro-fuzzy Systems. (3) An evaluation of the reviewed methods based on the performance under different wind conditions, stability system responses, and effectiveness modelling of uncertainties, training speed, computational power, and robustness to noise, as well as implementation complexity. (4) A discussion of the challenges and future directions of NFS-based MPPT methods for WECS.

II. OVERVIEW OF THE WIND ENERGY CONVERSION SYSTEMS

The wind energy conversion system is comprised of three main components, which are the blade rotor, gearbox, and electric generator [12]. The wind turbine's aerodynamics produce two low-pressure and high-pressure areas around the blades and spin them when the wind flows through. The turbine moves the electrical generator and converts wind energy to electricity [1], [3]. The block diagram of a typical grid-connected WECS is shown in Figure 1. Due to the nature of the wind, the WECS is categorized into two main classifications, namely fixed speed wind turbines (FSWT) and variable speed wind turbines (VSWT) [3]. The VSWTs inject significant harmonic currents into the grid while the FSWTs can alter the harmonic impedance of the distribution grid and, in some cases, create resonant circuits. However, there are serious problems facing

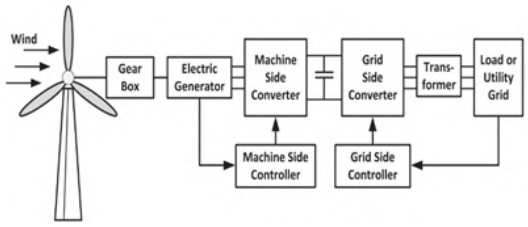


Fig. 1. Block diagram of a typical grid-connected WECS.

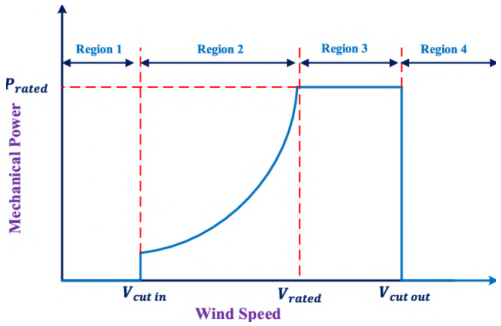


Fig. 2. Wind turbine operating regions [1].

FSWTs, such as restricted operating speed range, large mechanical stress, and a multiple-stage gearbox. Therefore, the VSWTs are aimed at overcoming these drawbacks, and they offer several advantages, including reduced power fluctuations and mechanical stress, enhanced efficiency through optimal power capture, and a higher degree of operational controllability [1] [3]. Furthermore, a wind turbine works within four operation zones as indicated in Figure 2, which represents the transmitted power of the wind turbine versus the wind speed variation. The wind turbine speed-control characteristics are essential to keep the turbine speed in a safe operating mode and to reduce the mechanical stress on the drivetrain [14]. Hence, a wind turbine speed-control strategy is used based on four operating regions, shown in Figure 2, which are described as follows:

- Region I: the wind speed is below the cut-in speed, too low to produce mechanical power,
- Region II: the wind speed lies between cut-in and rated speed. Furthermore, the wind turbine starts producing mechanical power. Also, the rotor speed of VSWTs is controlled to extract maximum power.
- Region III: the wind speed is above the rated wind speed but below the cut-out wind speed, the wind turbine is operated at the rated rotor speed to produce rated output power, and the wind power is enough to exceed the rated output power of the wind turbine.
- Region IV: When wind speed is above the cut-out wind speed, thus, the brakes are triggered to shut down the wind turbine and prevent any structural damage or unsafe conditions.

Wind power is generated with the help of an electrical generator (Fig 1), which is an electromechanical component

that converts mechanical power into electrical power, typically having a stator and a rotor [15]. The stator is a housing with mounted coils. The rotor has a rotating function that produces a magnetic field, which is induced in the windings of the stator, causing a voltage at the stator terminals. The rotor can be a permanent magnet or an electromagnet [12]. The most preferable options for VSWTs are the direct-drive Permanent Magnet Synchronous Generator (PMSG) and doubly fed induction generators (DFIG) [5]. Interest in PMSG is increasing due to their several benefits, such as enhanced efficiency, reliability, power density with improved performance, and requiring less maintenance [5].

A. Modelling of Wind Turbines

The power generation by the wind turbine is affected by the wind speed, the turbine rotor size, the blade swept area, the tip speed ratio, and the rotor speed. At a given wind speed, the mechanical power available to the shaft of a wind turbine is related to its rotation speed. The characteristic of power versus rotor speed is obtained by varying the wind speed, as shown in Figure 3.

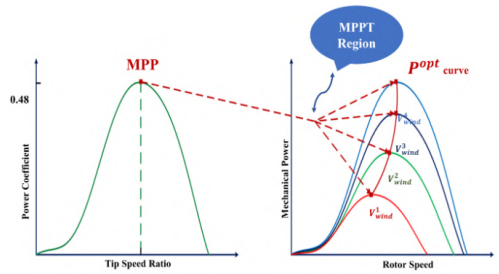


Fig. 3. Characteristics of wind turbine (WT): power coefficient (C_p) versus tip speed ratio (λ), and mechanical power versus rotor speed [2].

From a mechanical design point of view, wind turbines are designed to operate at maximum power for an average wind speed. However, the wind speed cannot stay at an average level all the time. For low wind speeds, the tip speed ratio increases (with a constant rotor speed), resulting in the reduction of the power coefficient. Thus, the wind turbine is unable to operate under optimal conditions most of the time [16]. Tracking the target operating point can be done by manipulating the power electronics and the appropriate regulators [2]. From the available energy possessed by the wind as shown in equations (1), (2), (3) and (4) only 59% is extracted by the WT by Betz limit [2]. Therefore, the power present in the air and the turbine power, P_{air} and P_{turbine}, are given by:

$$P_{\text{air}} = \frac{1}{2}\rho A V^3 \tag{1}$$

$$P_{\text{turbine}} = \frac{1}{2}\rho\pi R^2 C_p(\lambda,\beta)V_w^3 \tag{2}$$

where $\rho, A, V, C_p, \lambda, \beta$ represent the density of air (kg/m^3), area swept by the blades (m^2), wind speed (m/s), power coefficient, tip speed ratio (TSR), and blade pitch angle $(^\circ)$, respectively. The tip speed ratio λ is represented by Eq. (3) as follows:

$$\lambda = \frac{R\,\omega}{V} \tag{3}$$

where R is the radius (m), ω is the angular speed (rad/s), and V is the wind velocity (m/s). The air density is a variable parameter and can differ at distinct locations.

The power coefficient, denoted by C_p, is given by:

$$C_p = \frac{P_{\text{turbine}}}{P_{\text{air}}} \tag{4}$$

where P_{turbine} and P_{air} are the power absorbed by the wind turbine and the power present in the wind, respectively. The parameter C_p depends on the values of λ and the pitch angle β. Figure 3 portrays the relation between C_p and λ, illustrating the concept of maximum power point tracking (MPPT) for wind energy conversion systems (WECS).
Thus, C_p attains its highest value $C_{p,\text{max}} = 0.48$ for a particular value of λ [17], and this optimal value is useful in tracking the maximum power. Figure 3 also shows the relationship between the turbine mechanical power and the rotor speed of the electrical generator. Furthermore, the use of a permanent magnet synchronous generator has increased due to its reduced phase current, high power density with low torque ripples, and lower maintenance requirements [5] [18].

B. Permanent Magnet Synchronous Generator Modelling

As mentioned earlier, the generator converts mechanical power into usable electrical power. The dynamic model of a permanent magnet synchronous generator (PMSG) can be represented in the Park's (d-q) reference frame as follows [11]:

$$\frac{di_d}{dt} = \frac{-R_s i_d + p(L_q - L_{ch})\,\Omega_h i_q - R_{ch} i_d}{L_d + L_{ch}} \tag{4}$$

$$\frac{di_q}{dt} = \frac{-R_s i_q - p(L_q + L_{ch})\,\Omega_h i_d - R_{ch} i_q}{L_q + L_{ch}} + p\,\Omega_h \Phi \tag{5}$$

$$\frac{d\Omega_h}{dt} = \frac{1}{J_{HSS}}\left[-p\Phi i_q + \frac{d_1 v_w^2}{I} + \frac{d_2 v_w \Omega_h}{I^2} + \frac{d_3 \Omega_h^2}{I^3}\right] \tag{6}$$

where $R_s, L_d, L_q, p, \Omega_h, L_{ch}, \Phi, I, J_{HSS}, i_d$ and i_q represent the stator resistance, stator inductances along the d- and q-axes (for a non-salient pole PMSG, $L_d = L_q$), the number of pole pairs, the high-shaft angular speed, the load inductance, the constant flux linkage, the transmission ratio, the high-speed shaft inertia, and the stator currents, respectively.

The turbine-generated torque is expressed in Eqs. (7) and (9), showing the relationship between the electromagnetic torque (T_{em}), power (P_{em}), and rotor speed (Ω_{em}) as follows:

$$T_{em} = \frac{P_{em}}{\Omega_{em}} \tag{7}$$

The expression for the electromagnetic torque is given as:

$$T_{em} = \frac{3}{2} p \frac{L_m}{L_s}\left(\Phi_{qs} i_q - \Phi_{ds} i_d\right) \tag{8}$$

Then, considering the d-axis orientation, the electromagnetic torque developed by a PMSG can be deduced as follows:

$$T_{em} = -\frac{3}{2} p \frac{L_m}{L_s} \Phi_d i_d \tag{9}$$

With L_m and L_s are the mutual and leakage inductances, respectively.

III. DESCRIPTION OF MPPT IN WIND ENERGY CONVERSION SYSTEMS

The characteristics of a Wind Turbine (WT), as depicted in Figure 3, underscore the significance of the Maximum Power Point Tracking (MPPT) approach. The MPPT is a technique used to extract the maximum available power from a wind turbine. In the context of wind turbines, it can be used to control parameters such as pitch angle, the generator torque, duty cycle, voltage, wind speed, etc. [13]. Therefore, it is a crucial component of WECS, enabling them to operate at peak efficiency and generate the maximum power possible. The power output of a wind turbine does not linearly correlate with the wind speed. Instead, it follows a bell-shaped curve, peaking at a specific wind speed [18]. This peak, known as the MPP, varies based on wind speed and other factors [10]. MPPT methods are employed to track the MPP and maintain the wind turbine's operation at peak efficiency [6]. This can significantly enhance a wind turbine's energy output, particularly in low wind conditions. The fundamental principle of MPPT is to adjust the rotational speed of the turbine to match the optimal Tip-Speed Ratio (TSR) [8], [16] (the ratio of the blade tip speed to the wind speed) and the characteristics of the generator. The TSR varies with the wind speed, and there is a specific TSR value at which the wind turbine will generate maximum power. Therefore, it is essential to compel WT systems to operate at the MPP and extract the maximum power for every wind speed. This prevents the unique operating point MPP in the WT's characteristic curve from shifting its position when the wind speed changes. To select the appropriate MPPT method, parameters such as efficiency, speed, need for wind speed sensors, and implementation complexity are considered.

IV. REVIEW OF MPPT METHODS IN WIND ENERGY CONVERSION SYSTEMS

In the context of wind turbines, the MPPT process relies on the maximum power transfer theorem, which indicates that when the load resistance equals the source resistance, the maximum power transfer occurs [19]. In practical terms, this means that the MPPT controller adjusts the operating conditions of a wind turbine, such as its voltage and current, to ensure optimal power output. MPPT methods can be classified into two main types:

i. Conventional MPPT Methods: They are based on complex mathematical equations and require information about the wind turbine [19]. They require information about the wind turbine and are typically implemented using microcontrollers or digital signal processors [8].
ii. Intelligent MPPT Methods: They use AI techniques (through machine learning models) to track the maximum power point of the wind turbine [20] [21].
iii. Hybrid MPPT Methods: They are the combination of benefits of more than one method to reduce the limitations of individual methods, and it could be the mixture of conventional, conventional-intelligent, and/or intelligent methods [1] [2].

A. Conventional MPPT Methods

Conventional methods use complex mathematical equations and require information about the wind turbine [19], to calculate the maximum power point (MPP) of the wind power output. It is therefore challenging to measure the maximum power coefficient during wind speed variation because it is a nonlinear function of the tip-speed ratio (λ) and blade pitch angle. Thus, for a given blade pitch angle, there is a maximum value for the power coefficient C_p that corresponds to the optimal tip-speed ratio (λ) [8]. Different conventional MPPT methods have been proposed, and most fall into two categories: indirect power control (IPC)-based algorithms and direct power control (DPC)-based algorithms. The IPC enlarges the output power via regulating the mechanical power (P_m), according to prestored characteristics. The data of power-speed curves are used in the IPC approach, whereas the DPC analyses the alteration of output electrical power to track the maximum power point (MPP) [1]. The (TSR) control, optimum torque control (OTC), and power signal feedback (PSF) all fall under IPC, while optimum relation-based (ORB) control, incremental conductance, and perturbation and observation control fall under DPC [22] [8]. For instance, the conventional TSR controller regulates the optimal rotor speed under different environmental conditions and keeps the ratio between the tip of the blade and the rotor speed to an optimal value, where the extracted wind power is maximized, irrespective of variations in wind [6]. This method gives a fast response by using anemometers [8]. However, its performance depends on the wind speed information to compute the optimum value of the TSR, and the accuracy of wind speed is not guaranteed from direct measurements. Consequently, the installation cost increases [8]. Similarly, in an OTC method, the generator is controlled by the optimum torque reference curve derived from the maximum power of the WT, corresponding to the wind speed in time, and requires the air density parameter [8]. It is a fast method; however, it is less effective than TSR because the wind speed is not directly measured. Thus, wind changes are not instantly reflected in the set signal. Like TSR, the PSF method requires knowledge of the WT maximum power curve and depends on the optimal power (P_{opt}). The sensor-less PSF algorithms are implemented to estimate P_{opt}. However, their disadvantage is the need for a precise C_p and λ [1] [2]. Likewise, the ORB method relies on the predetermined relationship between the parameters of the WT. It offers rapid response and simple operation without anemometers. The simplicity of this method lies in the fact that only the values of DC power, voltage, and current serve as the inputs [2]. Nevertheless, high accuracy,

TABLE I
SUMMARY OF REFERENCES OF CONVENTIONAL MPPT METHODS

Type of Method	Remarks
TSR [14]	TSR regulates the optimal rotor speed under different wind conditions and keeps the ratio between the tip of the blade and the rotor speed at an optimal value, where the extracted wind power is maximized irrespective of variations in wind speed.
PSF [1] [2]	PSF estimates the optimal power by comparing dc-link change and current wind speed, and it efficiently estimates the rotor speed reference by detecting optimal power curves and measured electrical power.
ORB [2] [1]	The ORB uses optimal relations such as power versus rotor speed, electromagnetic torque, or power versus rectified dc-link voltage (or dc-link voltage and current) for rapid response and simple operation.
OTC [24]	OTC controls the optimum torque reference curve, derived from the maximum power of the wind turbine corresponding to wind speed over time.
P&O [1]	This method does not require prior knowledge of the optimum point characteristic curve and is suitable for low-inertia wind turbine systems.
INC [25]	The use of sensors and knowledge of wind turbine parameters is eliminated. INC involves the observation of rectifier output power to track the maximum power point and determine the perturbation direction.

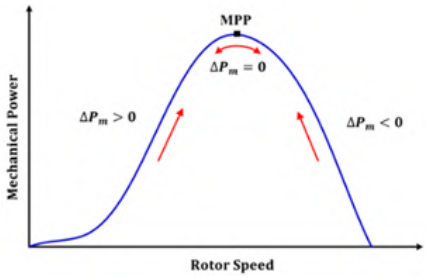


Fig. 4. Operation concept of P&O, adopted from [1] [2].

field tests, and high memory are required for each WECS to obtain the optimum relationship [23]. Table I summarizes the conventional methods and remarks on their differences.

On the contrary, the incremental conductance (INC) method involves the observation of the rectifier output power and the calculation of the power variation slope, similar to the perturbation and observation (P&O) method. Thus, varying the perturbation of the DC-link voltage (V_{dc}) step sizes enhances the overall dynamic performance of the wind energy system and improves the convergence speed. The need for sensors and knowledge of the WT parameters is eliminated by using the INC method, resulting in increased reliability and efficiency [2]. However, it has a slower response time, which reduces efficiency. Equally, the P&O control is one of the most commonly used algorithms because of its flexibility and simplicity. The approach for the P&O algorithm involves locating the MPP by employing a mathematical optimization strategy. It does not require any prior knowledge of the WT; rather, a hill-climb search (HCS) technique is used to locate the local optimum power [2]. Figure 4 shows the operational concept of the MPP obtained when the variation in mechanical power is null. Moreover, the P&O algorithm perturbs the control variables, such as rotor speed or DC-link voltage, and observes their influence on the WECS. It does not require sensors such as an anemometer. Figure 5 illustrates the flow chart of the P&O algorithm used to track the MPP. The algorithm perturbs the control variables with a precise step size and detects the goal function until the MPP is achieved. The obtained power is typically adjusted along with the generator speed to achieve a zero slope on the power versus rotor speed curve, as shown in Figure 4. However, the P&O algorithm employs a fixed step size, which may fail when applied to large WTs under varying wind speeds. Consequently, oscillations occur near the MPP [23].

B. Intelligent MPPT Methods: ANN and Fuzzy logic

Intelligent MPPT methods use soft computer-based reasoning and machine learning models to track the maximum power point of the wind turbine [20] [21] [26]. Intelligent methods such as ANN, and fuzzy systems (FLS). [20] [21] [26] are developed to overcome these challenges of the conventional methods and this includes the need for the accurate mathematical model, wind speed information, slower response time, oscillations near MPP, etc., This section presents Fuzzy-Logic methods, ANN methods and their hybrids.

1) The fuzzy logic (FL) method: FL is an intelligent computing approach based on degrees of truth instead of the usual true or false Boolean logic on which modern computer systems are based [7]. The ability to quickly adjust the controller variables distinguishes this nonlinear method, corresponding to the changes that occur in the dynamics of the system [7] [23]. It also allows the manipulation of imprecise or uncertain data [13], and the information needed about the fuzzy rules is obtained with the knowledge of an expert on the system. FL steps, described in Figure 5 include fuzzification, fuzzy inference engine, and defuzzification [23]. FL has advantages such as flexible operation, a convenient user interface, and ease of implementation. Ngo et al [8] designed an MPPT controller using FL for small-scale WT. To overcome the drawback of P&O same as in [27], the proposed algorithm created an appropriate step-size from WT power and the generator speed variables. It gives the optimal step for the system, and the results showed better system response because it reflects the actual wind speed at the control time, and the power in the range of low wind speed is exploited with the optimal value. Benaaouinate [18] applied FLC to develop an MPPT method without using characteristic data provided by the manufacturer for a given WT to maximize the energy extracted from the wind generator, and to control the grid-tied inverter for DC-Link voltage regulation, substituting the proportional-integral (PI) controller to keep the DC-Link voltage constant. Similarly, an efficient control system that depends on the adaptive FLC (AFLC) approach to obtain the MPP under different wind speeds was proposed by Salem [28]. As a result, they reported a better tracking capability and a reduction of oscillation rate when compared to simple FL.

Furthermore, the parameters of the fuzzy controller can be rapidly adjusted in response to system dynamics, and when the climate conditions change, the performance of a fuzzy-based MPPT method is strong. However, its effectiveness depends a lot on the user's knowledge in selecting the appropriate error, membership functions, and rule base selection. Moreover, the memory required imposes constraints on its implementation.

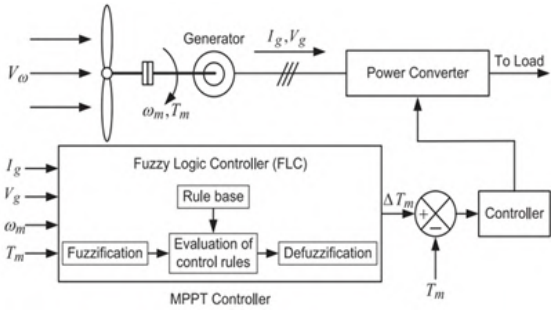


Fig. 5. Fuzzy-based MPPT method of WECS [11].

2) Artificial Neural Networks methods: ANNs are a machine learning model with a non-linear computational model inspired by the neural networks of living creatures and enable the design of complex systems. It regulates the optimal condition of different control variables along with the inputs and appropriate weights of layers in a WT. It is capable of decision-making, learning, and modelling with flexibility [29]. The ANN model utilizes a variety of input variables, including terminal voltage, wind speed, and output torque, among others. The model's architecture primarily consists of an input layer, one or more hidden layers, and an output layer, as shown in Figure 6, each populated with neurons. The optimal number of neurons in the hidden layers is determined through a process of trial and error [29]. The model's output, which could be a reference power or rotor speed, guides the operation of the power electronic circuitry. The weights realised in the layers decide how the operating point will coincide with the peak point. Then the training of the system results in the reduction of the response time, which facilitates the quick achievement of the steady state even under varying wind velocities. Elbej et al [9] designed an ANN to estimate the PMSG speed and calculate the error between the estimated and the desired output. Mustapha [30] proposed an ANN-based MPPT that utilizes a Nonlinear Autoregressive with External Input based on neural networks to extract the maximum available power from different wind velocities. The trained ANN was able to predict optimum values for the duty cycle to extract maximum power.
A new MPPT-optimal torque control (MPPT-OTC) is proposed by Meghni [31], and the ANN regulator developed eliminates the disadvantages of the OTC-MPPT method on a PI controller. The OTC-ANN shows

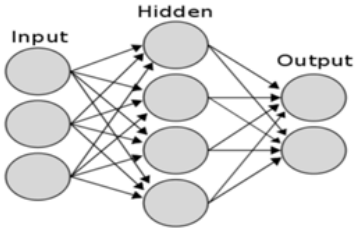


Fig. 6. Structure of ANN used for estimation of the variables [11].

a faster response time and robustness in wind gusts compared with OTC-PI. Rahman [16] employed an adaptive network-based radial basis function algorithm to track the optimum rotor conditions of the generator under changing wind speed. The ANN rapidly adjusts the generator-side converter reference speed settings, and the maximum power could be extracted from the WT with an optimum power coefficient for changing conditions. The efficiency of the ANN controller is generally good since most mechanical parts have varying characteristics with aging and under different environmental conditions. However, the neural network must be periodically trained to guarantee accurate MPPT, and a large memory size is required [11]. Table II compares the characteristics of intelligent and conventional methods [2].

C. Hybrid MPPT Methods

The principle of hybrid MPPT methods is mainly based on merging various features of different MPPT methods to reduce the limitations of individual methods. Table II summarizes different intelligent hybrid methods and remarks on their differences. Some of the most promising hybrid techniques are based on neuro-fuzzy systems. Neuro-fuzzy systems are a hybrid intelligent control system that combines the learning and adaptation capabilities of artificial neural networks with the human-like reasoning capabilities of fuzzy logic. In the context of WECS, an ANFIS can be used to control various aspects of a wind turbine, such as the pitch angle of the blades, the yaw angle of the nacelle, and the generator torque [13]. The goal of the NFIS is to optimize the performance of the wind turbine while also ensuring safe operation. Neuro-fuzzy systems have been used in a wide range of applications, including MPPT systems, and the most used for MPPT in WECS include adaptive neuro-fuzzy inference systems; type-2 fuzzy neural network systems; optimized neuro-fuzzy systems; and wavelet neuro-fuzzy systems.

1) *Adaptive Neuro-Fuzzy Inference System*: ANFIS is a type of NFS that uses a supervised learning algorithm to train the fuzzy model. ANFIS has been used to develop MPPT controllers for WECS with different types of wind turbines and generators, such as PMSGs and DFIGs. In ANFIS, the learning algorithm is hybrid. It works by combining back-propagation, gradient descent, and a least-squares algorithm to identify and optimize the Sugeno system parameters [32]. Some

of the advantages of ANFIS are to model nonlinear and complex functions, identify nonlinear components online, and predict a messy time series [32]. Ahmed [4] proposed an ANFIS controller for grid-connected DFIG that extracts maximum power from the wind by tracking the MPP regardless of wind speed. An ANFIS was proposed by Asghar and Liu [15] for online estimation of effective wind speed from the instantaneous values of WT, tip speed ratio, rotor speed, and mechanical power. As a result, the estimation accuracy of the ANFIS is observed for effective wind speed data in region 2. An adaptive controller of a microgrid using an ANFIS was utilized by Abdalla [26] to regulate the load voltage and frequency. The ANFIS controller gave greater reliability in terms of power generation and distribution compared to simple FL. In Phan's work [21], a Wind Speed Estimator-Based Output Feedback Near-Optimal MPPT was proposed. The ANFIS parameters are updated using the hybrid learning rule to reduce the calculation burden. Consequently, the simulation shows that the ANFIS estimator has fast dynamics even under changing conditions. Nevertheless, simple fuzzy systems are complex due to the absence of certain rules for determining membership function values and linguistic variables.

2) *Type-2 Fuzzy Neural Networks:* Type-2 fuzzy neural networks (T2FNNs) arise from the absence of certain rules in determining the exact values of membership functions and linguistic variables in type-1 fuzzy, leading to uncertain situations [33]. Type-2 fuzzy method produces more stable system responses due to effective modelling of uncertainties. Therefore, it extends the capabilities of traditional fuzzy logic by allowing for the modelling of uncertainties at a higher level [34]. T2FNNs incorporate this idea into NN architectures, providing more robust handling of both input and rule parameters [33]. T2FNN deals with uncertain and imprecise information by considering not only the membership degrees but also the uncertainty in those degrees and adjusting the parameters of the fuzzy logic system and the artificial neural network [13]. The ANN component learns and adapts to the complex and nonlinear relationships between input variables (like wind speed, turbine characteristics) and output variables (optimal power point) [13] [34]. Moradi et al. [35] proposed an adaptive sliding mode T2FNNs controller for power control of DFIG. The parameters of T2FNNs are tuned by the Sliding Mode control-based methods. The results show that the transient response of the overall system is improved, and it indicates that the proposed controller reduces the power errors considerably. However, to achieve more satisfactory performance while dealing with uncertain information or data generated from uncertain sources (uncertainties even in the presence of uncertainty), research work on higher-order fuzzy systems is being explored. Furthermore, uncertainties in uncertainty indicate that there exists a high degree of vagueness in data or information, and to handle such a system, higher types of FSs are needed.

3) *Wavelet Neuro-Fuzzy Systems:* The Wavelet neuro-fuzzy system uses wavelet networks and fuzzy logic to develop a system that has a fast-training speed, increases the computational power, and can describe nonlinear objects that are characterized by uncertainty [36]. In the work of Hassan et al [37], a neuro-fuzzy indirect wavelet-based adaptive MPPT control is developed to ensure fast response and high efficiency under different weather conditions. Hong [36] proposed an Enhanced Radial Fuzzy Wavelet Neural Network (RFWNN) with HCS MPPT for a wind turbine generator. The Enhanced RFWNN combines the TSK fuzzy with wavelet functions to increase the computational power of the NFS, and the proposed method gives better results than fuzzy and PI methods. Muhammad et al. [38] used an NF Full-Recurrent Hybrid B-Spline Wavelet-Based Feedback Linearization Control for wind turbines. The method estimates the nonlinear functions of a Feedback Linearization using a B-Spline membership function and the Morlet wavelet network in the antecedent and the consequent parts, respectively. The proposed method outperformed the PID in terms of reference tracking, power quality, and power system stability for highly nonlinear wind speed profiles. However, computational complexity is still a burden because some wavelet algorithms require significant computational resources, especially for real-time applications, which can be challenging for onboard wind turbine systems.

4) *Optimized Neuro-Fuzzy Systems:* Optimized models enhance the power extraction by minimizing objective functions. Though most neuro-fuzzy systems can manage non-linearity without an exact mathematical model and supply conspicuous tracking efficiency. However, they require a large quantity of data for their training process, putting a strain on processor memory [32]. Thus, optimized neuro-fuzzy systems combine two or more optimization methods, which are also intelligent models, to find the parameters and achieve stable and fast convergence [39]. It may integrate genetic algorithms (GA), particle swarm optimization (PSO), or other metaheuristic algorithms to fine-tune the neuro-fuzzy model for improved MPPT [39]. Elimany et al. [40] proposed a new hybrid MPPT technique including zebra optimization algorithm (ZOA)-ANFIS-based MPPT and gorilla troops optimizer (GTO)-ANFIS-based MPPT to obtain the maximum power of wind turbines and solar panels. The two introduced methods evaluated the instantaneous maximum power extraction from both energy sources. The hybrid GTO-ANFIS-based MPPT method integrates the GTO algorithm while the ANFIS optimizes the control parameters of the wind energy system, accurately tracking its MPP. The results show that ZOA outperforms the GTO method in the computation time. In [20], a hybrid ANFIS-GA-based control

TABLE II
SUMMARY OF REFERENCES OF INTELLIGENT/HYBRID MPPT METHODS

Type of Method	**Simulation Environment**	Remarks
OTC-ANN [31]	MATLAB/ Simulink	The generator torque is controlled to obtain the maximum value of the power coefficient.
FL-P&O, FL-PI [27] [43]	MATLAB/ Simulink	In [27] FL selects an appropriate step size to resolve difficulties within the P&O. In [43] FL-PI tunes the duty cycle of the boost converter.
ANFIS, ANFIS-PID [20] [15]	MATLAB/ Simulink	ANFIS estimates the effective wind speed. The ANNs train fuzzy MFs to adopt the system behaviour.
T2FNNs [44] [35] [33]	MATLAB/ Simulink	Interval T2FS is used to handle uncertainty in wind speed and system parameters. SMC adapted antecedent and consequent T2FS parameters based on inputs (active /reactive power errors, rotor speed errors, and their derivatives).
RFWNN [37]	MATLAB/ Simulink	The HWNF gradient-descent-based back propagation algorithm is developed to update the parameters of the proposed structure adaptively. The combination allows for to construction of a system with a rapid training speed and to describe nonlinear objects with uncertainty.
ANFIS-GA [15]	MATLAB/ Simulink	ANFIS-GA algorithm controls both the converter and the inverter, while a GA-based learning design identifies the ANFIS parameters.
ZOA-ANFIS & GTO-ANFIS [40]	MATLAB/ Simulink	The hybrid ZOA-ANFIS-based MPPT technique employs the Zone of artificial ZOA to continuously adapt the operating voltage of the PV system for maximum power point tracking. The hybrid GTO-ANFIS-based MPPT technique integrates the GTO algorithm while the ANFIS optimizes the control parameters of the wind energy system, accurately tracking its maximum power point.
ANFIS-PSO [41]	MATLAB/ Simulink	The PSO offers a simpler, faster, and less complex method for training and updating ANFIS specifications without initial parameter calculation or learning rate requirements.

is employed to enhance the performance of a WECS. ANFIS is used to fine-tune the MFs of the FIS to minimise the error, while GA learning design procedure is proposed to identify the ANFIS parameters. The performance of the controller is evaluated by using both dynamic and transient analyses. It is claimed from the simulation results that the dynamic and transient system responses of ANFIS-GA control strategy have better damped and faster responses with minimum oscillations than those of the GWO-based optimised PI control strategy. Priyadarshi in [41] introduces an adaptive NFIS-particle swarm optimization (ANFIS-PSO)-based hybrid MPPT method to acquire rapid and maximal power with zero oscillation tracking. The comparison with other models shows that PSO has better PV power tracking ability. However, some optimized algorithms may not track the global MPP in a dynamic state [42]. Thus, the challenge of choosing an appropriate optimizer.

V. COMPARATIVE ANALYSIS AND DISCUSSION

A comparative analysis was conducted on various MPPT methods, with a focus on performance under wind variation, wind speed measurements, stability of the system responses, and effectiveness modelling of uncertainties, training and convergence speed, computational power or memory requirement, the robustness to noise, as well as the implementation complexity. The primary objective of the MPPT method is to track the optimal power point of a variable-speed wind turbine, making the selection of an appropriate MPPT technique a challenging task. Indirect power control-based methods, such as Tip-Speed Ratio Optimal Tracking and Power Signal Feedback, are simple and fast. They, however, maximize the captured mechanical wind power rather than the output electrical power, affecting the overall performance. Direct power-based control techniques, including P&O, INC, and ORB, are sensorless, straightforward, and require less memory. These techniques directly calculate optimal electrical power without the need for prior training or wind speed measurement. Despite their simplicity, their performance is unsatisfactory under wind variation, limiting their application under varying wind conditions. Nevertheless, most of the shortcomings of the conventional methods can be addressed through intelligent methods, algorithm modification, or a hybrid approach. Most intelligent methods have a high convergence speed and do not require complex mathematical equations or wind speed measurements to estimate the power output of the wind turbines. For instance, intelligent MPPT methods, based on ANN, fuzzy logic, and NFS, efficiently predict the optimal power and work well with nonlinear systems, indicating good performance under wind variation. Both FL and NN-based MPPT controllers offer a better balance in terms of system dynamic speed, leading to the stability of the system responses and power responses compared to conventional methods. The efficiency of the NN controller is generally high, even under different environmental conditions. Nevertheless, to ensure accurate MPPT, the neural network must be trained periodically, and a large memory requirement is necessary. In fuzzy logic systems, the processing time is influenced by the number of rules present in the controller. In other words, the more rules a fuzzy logic system has, the longer it takes to compute the desired output. This consideration is essential when designing and optimizing such systems. Neuro-Fuzzy systems MPPT methods, such as adaptive NFIS; T2FNNs, optimized NFS, and wavelet neuro-fuzzy systems, hold significant advantages over simple intelligent methods, as they integrate the human-like reasoning capabilities of fuzzy logic with adaptation capabilities of artificial neural networks to address the oscillations around the MPP by dynamically adjusting the parameters of fuzzy membership functions. As a result, they exhibit enhanced adaptability, robustness, and performance accuracy, particularly under wind speed variation and power demand. Similarly, interval Type-2 Neuro-fuzzy Systems are designed to augment the capabilities of traditional fuzzy logic. They enhance the modelling of uncertainties

by integrating membership degrees within neural network structures. Additionally, they adapt to complex relationships and enhance the transient response while minimizing power errors. On the other hand, the Wavelet NFS is employed to create a system with rapid training speed and the ability to describe nonlinear objects with uncertainty. Moreover, in the pursuit of addressing uncertainties more flexibly, researchers are investigating general type-2 and type-3 fuzzy systems as extensions of type-2. While type-2 systems account for membership function uncertainties, general type-2 and type-3 systems allow for non-uniform weighting of these uncertainties. These advanced fuzzy systems will offer improved modelling capabilities, especially with complex and varied uncertainties in WECS applications. Equally, Optimized NF systems have the advantage of combining multiple optimization algorithms, which help to improve the global search capability to track the real MPP, to achieve a stable convergence speed. Though optimized algorithms improve the search capability of the MPPT controller in a dynamic state, some optimized algorithms, such as GA, may not be recommended to optimize at a very large scale, because of highly complex and excessive problems due to its simplified algorithm, which may lead to a failure in tracking the global MPP. However, the downside of neuro-fuzzy methods may lie in acquiring a substantial volume of data for training machine learning models. This requirement places significant demand on processing memory and extends the time needed for training and learning. But understanding the trade-offs between data availability and model performance is crucial in machine learning endeavours.

VI. CHALLENGES AND FUTURE DIRECTIONS

MPPT in wind energy systems is a fundamental building block for maximizing the efficiency and performance of the systems. The focus on optimizing this technique has generated significant interest in the research community, and Neuro-fuzzy-based techniques have emerged as a key component in this field. Neuro-fuzzy has been shown to have the highest efficiency, fastest convergence, and high stability in terms of oscillation around the MPP. However, by compiling these publications, we detected the following, but not limited to, relevant open directions and several challenges: (1) Proposing new hybrid approaches. The objective is to lower the computational burden/ memory of hardware and improve their performance by increasing the tracking efficiency. Frequently, combining offline and online techniques can lead to enhancing the performance of the whole system. (2) Developing new approaches to increase the training speed of the model. (3) Testing new metaheuristics approaches for global MPP tracking. (4) Developing methodologies, guidelines, and/or selection criteria to help the designers of the WECS choose the best MPPT [23] [36]. (5) Exploring hybrid renewable energy systems by incorporating distributed generation sources like solar, fuel cells, biomass, and hybrid electric vehicles [34].

VII. CONCLUSION

Wind Wind energy conversion systems represent a highly promising power source due to their abundance and appeal among renewable energy options. Nevertheless, these systems experience inherent nonlinearity from stochastic losses, which can destabilize power output. Because wind turbine efficiency depends greatly on effective control strategies, a range of machine learning models have been integrated into their control systems. Notably, the use of machine learning approaches for MPPT controllers has played a crucial role in maximizing energy capture from wind energy conversion systems. This investigation highlights the diverse array of machine-learning-based MPPT techniques now available, including individual methods such as ANN, fuzzy logic, P&O, and INC, as well as hybrid methods like ANFIS, ANN-P&O, FL-P&O, and ANFIS-GA. Many of these hybrid methods consistently outperform traditional MPPT techniques. Neuro-fuzzy hybrid systems demonstrate strong potential, delivering better results than single-method approaches. Type-2 fuzzy neural networks offer improvements over classical fuzzy logic by handling model complexity, while wavelet neuro-fuzzy systems accelerate training speed. Optimized neuro-fuzzy approaches also enable more effective global maximum searches, supporting rapid and stable MPP tracking. As a result, hybrid MPPT methods stand out as a promising research direction for developing robust wind turbine power conversion systems.

REFERENCES

[1] H. H. H. Mousa, A. R. Youssef, and E. E. M. Mohamed, "State of the art perturb and observe mppt algorithms based wind energy conversion systems: A technology review," *International Journal of Electrical Power and Energy Systems*, vol. 126, no. PA, p. 106598, 2021.

[2] J. Pande, P. Nasikkar, K. Kotecha, and V. Varadarajan, "A review of maximum power point tracking algorithms for wind energy conversion systems," *Journal of Marine Science and Engineering*, vol. 9, no. 11, p. 1187, 2021.

[3] M. Alzayed, H. Chaoui, and Y. Farajpour, "Maximum power tracking for a wind energy conversion system using cascade-forward neural networks," *IEEE Transactions on Sustainable Energy*, vol. 12, no. 4, pp. 2367–2377, 2021.

[4] A. A. Chhipa *et al.*, "Adaptive neuro-fuzzy inference system-based maximum power tracking controller for variable speed wecs," *Energies*, vol. 14, no. 19, p. 6275, 2021.

[5] M. Abolvafaei and S. Ganjefar, "Maximum power extraction from wind energy system using homotopy singular perturbation and fast terminal sliding mode method," *Renewable Energy*, vol. 148, pp. 611–626, 2020.

[6] O. Apata and D. T. O. Oyedokun, "An overview of control techniques for wind turbine systems," *Scientific African*, vol. 8, p. e00566, 2020.

[7] D. R. Lopez-Flores, J. L. Duran-Gomez, and M. I. Chacon-Murguia, "A mechanical sensorless mppt algorithm for a wind energy conversion system based on a modular multilayer perceptron and a processor-in-the-loop approach," *Electric Power Systems Research*, vol. 186, p. 106409, Sep. 2020.

[8] Q. V. Ngo, C. Yi, and T. T. Nguyen, "The maximum power point tracking based-control system for small-scale wind turbine using fuzzy logic," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 4, pp. 3927–3935, 2020.

[9] O. Elbeji, M. Hannachi, M. Benhamed, and L. Sbita, "Artificial neural network–based sensorless control of wind energy conversion system driving a permanent magnet synchronous generator," *Journal of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 45, no. 3, pp. 459–476, Feb. 2020.

[10] D. A. Umar *et al.*, "Evaluating the efficacy of intelligent methods for maximum power point tracking in wind energy harvesting systems," *Processes*, vol. 11, no. 5, 2023.

[11] M. A. Khan, Q. Khan, L. Khan, I. Khan, A. A. Alahmadi, and N. Ullah, "Robust differentiator-based neurofuzzy sliding mode control strategies for pmsg-wecs," *Energies*, vol. 15, no. 19, pp. 1–18, 2022.

[12] E. C. Navarrete, M. T. Perea, J. C. J. Correa, R. V. C. Serrano, and G. J. R. Moreno, "Expert control systems implemented in a pitch control of wind turbine: A review," *IEEE Access*, vol. 7, pp. 13 241–13 259, 2019.

[13] D. Belkhiri, M. Ajaamoum, K. Cherifi, A. Elidrissi, and M. E. R. M. Alaoui, "Artificial intelligence-based mppt techniques in wind energy systems: A literature review," in *2023 International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, 2023.

[14] G. B. A. Kumar and Shivashankar, "Optimal power point tracking of solar and wind energy in a hybrid wind solar energy system," *International Journal of Energy and Environmental Engineering*, vol. 13, no. 1, pp. 77–103, Mar. 2022.

[15] A. B. Asghar and X. Liu, "Adaptive neuro-fuzzy algorithm to estimate effective wind speed and optimal rotor speed for variable-speed wind turbine," *Neurocomputing*, vol. 272, pp. 495–504, Jan. 2018.

[16] M. M. A. Rahman and A. H. M. A. Rahim, "Design and testing of an mppt algorithm using an intelligent rbf neural network and optimum relation based strategy," in *2020 IEEE Region 10 Symposium (TENSYMP)*, 2020, pp. 1245–1248.

[17] P. Gajewski, "Control of the hybrid renewable energy system with wind turbine, photovoltaic panels and battery energy storage," *Energies*, 2021, available at mdpi.com.

[18] L. Benaaouinate, M. Khafallah, D. Voyer, A. Mesbahi, and T. Bouragba, "Nonlinear control based on fuzzy logic for a wind energy conversion system connected to the grid," *International Journal of Renewable Energy Research*, vol. 10, no. 1, pp. 193–204, 2020.

[19] M. A. Mossa, H. Echeikh, and A. Iqbal, "Enhanced control technique for a sensor-less wind driven doubly fed induction generator for energy conversion purpose," *Energy Reports*, vol. 7, pp. 5815–5833, Nov. 2021.

[20] M. A. Soliman, H. M. Hasanien, H. Z. Azazi, E. E. El-kholy, and S. A. Mahmoud, "Hybrid anfis-ga-based control scheme for performance enhancement of a grid-connected wind generator," *IET Renewable Power Generation*, vol. 12, no. 7, pp. 832–843, May 2018.

[21] N. M. L. Phan, D. N. Tung, T. N. Thanh, and N. T. T. Vu, "Anfis wind speed estimator-based output feedback near-optimal mppt control for pmsg wind turbine," *Journal of Control, Automation and Electrical Systems*, vol. 34, no. 3, pp. 588–598, Jun. 2022.

[22] J. G. Njiri and D. Söffker, "State-of-the-art in wind turbine control: Trends and challenges," *Renewable and Sustainable Energy Reviews*, vol. 60, pp. 377–393, 2016.

[23] T. George, P. Jayaprakash, U. Subramaniam, and D. J. Almakhles, "Frame-angle controlled wavelet modulated inverter and self-recurrent wavelet neural network-based maximum power point tracking for wind energy conversion system," *IEEE Access*, vol. 8, pp. 171 373–171 386, 2020.

[24] F. Merahi and H. Mernache, "Optimal torque control for power optimization in pmsg based wind energy conversion system for an autonomus mode operation," in *2022 International Conference on Signals, Systems Devices (SSD)*, 2022.

[25] H. E. aissaoui, A. E. ougli, and B. Tidhaf, "A new maximum power point tracking based on neural networks and incremental conductance for wind energy conversion system," in *E3S Web of Conferences*, vol. 336, 2022.

[26] S. A. Abdalla, S. S. Abdullah, and A. M. Kassem, "Performance enhancement and power management strategy of an autonomous hybrid fuel cell/wind power system based on adaptive neuro fuzzy inference system," *Ain Shams Engineering Journal*, vol. 13, no. 4, p. 101655, Jun. 2022.

[27] K. Chaicharoenaudomrung, N. K. Areerak, N. K. Areerak, N. S. Bozhko, and N. C. I. Hill, "Maximum power point tracking for stand-alone wind energy conversion system using flc-po method," *IET Electric Power Applications*, vol. 15, no. 12, pp. 1723–1733, Dec. 2020.

[28] A. A. Salem, N. A. N. Aldin, A. M. Azmy, and W. S. E. Abdellatif, "Implementation and validation of an adaptive fuzzy logic controller for mppt of pmsg-based wind turbines," *IEEE Access*, vol. 9, pp. 165 690–165 707, 2021.

[29] B. C. Phan and Y. C. Lai, "Control strategy of a hybrid renewable energy system based on reinforcement learning approach for an isolated microgrid," *Applied Sciences*, vol. 9, no. 19, 2019.

[30] A. Mustapha, S. K. Selvaperumal, H. Mohd, and R. Lakshmanan, "Ann-based maximum power point tracking of a variable-speed wind energy conversion system using sepic converter," *International Journal of Advanced Science and Technology*, vol. 29, no. 1, pp. 189–205, 2020.

[31] B. Meghni, H. Chojaa, and A. Boulmaiz, "An optimal torque control based on intelligent tracking range (mppt-otc-ann) for permanent magnet direct drive wecs," in *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, 2020.

[32] M. I. S. Guerra, F. M. U. de Araújo, J. T. de Carvalho Neto, and R. G. Vieira, "Survey on adaptative neural fuzzy inference system (anfis) architecture applied to photovoltaic systems," *Energy Systems*, 2022.

[33] A. Gani, "Advanced control of positive output self-lift luo converter using type-2 fuzzy neural network via elliptic membership function," in *2023 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, Oct. 2023, pp. 1–6.

[34] B. Pakkiraiah and G. D. Sukumar, "Performance and analysis of an asynchronous motor drive with a new modified type-2 neuro fuzzy based mppt controller under variable irradiance and variable temperature," *International Journal of Computers and Applications*, vol. 38, no. 2–3, pp. 61–74, 2017.

[35] H. Moradi, Y. Alinejad-Beromi, H. Yaghobi, and D. Bustan, "Sliding mode type-2 neuro-fuzzy power control of grid-connected dfig for wind energy conversion system," *IET Renewable Power Generation*, vol. 13, no. 13, pp. 2435–2442, 2019.

[36] C. M. Hong and C. H. Chen, "Enhanced radial fuzzy wavelet neural network with sliding mode control for a switched reluctance wind turbine distributed generation system," *Engineering Optimization*, vol. 51, no. 7, pp. 1133–1151, 2019.

[37] S. Z. Hassan *et al.*, "Neuro-fuzzy wavelet based adaptive mppt algorithm for photovoltaic systems," *Energies*, vol. 10, no. 3, p. 394, 2017.

[38] M. Awais, L. Khan, R. Badar, S. Ahmad, S. Mumtaz, and S. Ullah, "Neurofuzzy full-recurrent hybrid b-spline wavelet based feedback linearization control for pmsg-wecs in a grid-connected hybrid power system," *IEEE Systems Journal*, 2021.

[39] K. V. Shihabudheen and G. N. Pillai, "Recent advances in neuro-fuzzy system: A survey," *Knowledge-Based Systems*, vol. 152, pp. 136–162, 2018.

[40] M. M. Elymany, M. A. Enany, and N. A. Elsonbaty, "Hybrid optimized-anfis based mppt for hybrid microgrid using zebra optimization algorithm and artificial gorilla troops optimizer," *Energy Conversion and Management*, vol. 299, no. August 2023, p. 117809, 2024.

[41] N. Priyadarshi *et al.*, "An experimental estimation of hybrid anfis-pso-based mppt for pv grid integration under fluctuating sun irradiance," *IEEE Systems Journal*, vol. 14, no. 1, 2020.

[42] M. M. Rahman and M. S. Islam, "Pso and ann based hybrid mppt algorithm for photovoltaic array under partial shading condition," *Engineering International*, vol. 8, no. 1, pp. 9–24, 2020.

[43] F. E. Tahiri, K. Chikh, and M. Khafallah, "Designing a fuzzy-pi controller of a stand-alone wind energy conversion system for mppt," in *Advances in Intelligent Systems and Computing*, 2019, pp. 1093–1106.

[44] H. Moradi, H. Yaghobi, Y. Alinejad-Beromi, and D. Bustan, "Power-control and speed-control modes of a dfig using adaptive sliding mode type-2 neuro-fuzzy for wind energy conversion system," *IET Renewable Power Generation*, vol. 14, pp. 2946–2954, 2020.

Tiny-CNN: Achieving State-of-the-Art Solar Panel Soiling Detection on Edge Devices through Knowledge Distillation and Quantization

Tyron Ncube
Department of Computer Science
Rhodes University
Makhanda, South Africa
tyronncube@gmail.com

Nomusa Dlodlo
Department of Computer Science
Rhodes University
Makhanda, South Africa
n.dlodlo@ru.ac.za

Josephat Kalezhi
School of ICT
Copperbelt University
Kitwe, Zambia
kalezhi@cbu.ac.zm

Alfredo Terzoli
Department of Computer Science
Rhodes University
Makhanda, South Africa
a.terzoli@ru.ac.za

Abstract— **The adoption of renewable energy, particularly solar energy, has increased over the past few years. Still, the operational efficiency of photovoltaic systems is severely undermined by soiling, necessitating automated, on-site monitoring solutions. While deep learning models offer high accuracy for soiling detection, they typically face a critical trade-off between state-of-the-art architectures that are too computationally expensive for low-cost edge devices and lightweight models that suffer from compromised performance. This paper resolves this challenge by introducing Tiny-CNN, a custom-designed, ultra-lightweight convolutional neural network optimized through a novel two-stage pipeline. We first employ knowledge distillation to transfer the rich representations from a large ResNet50v2 teacher model to boost our model's accuracy. We then apply post-training dynamic range quantization to enhance its on-device performance further. Evaluated on a Raspberry Pi 5, our final model achieved a state-of-the-art classification accuracy of 92.27%, surpassing all benchmark models tested, including its heavyweight teacher. More remarkably, it achieved this with a real-time inference speed of 18.18 ms per image and a final model size of only 119 KB. Our work demonstrates a field-ready solution that eliminates the compromise between accuracy and efficiency and presents a practical blueprint for developing intelligent edge AI systems for the renewable energy industry.**

Keywords— **Edge AI, Knowledge distillation, Quantization, ICT4D**

I. Introduction

The global transition towards renewable energy sources has positioned solar power as a cornerstone of sustainable energy production. However, the efficiency of these photovoltaic (PV) systems is significantly hindered by soiling. Soiling in this context can be defined as the accumulation of dust, dirt, and other particulates on the surface of a solar panel. This soiling can lead to substantial power losses that can exceed 50% in the dustiest regions [1], and managing this requires costly and labour-intensive maintenance schedules. For large-scale solar farms, manual inspections are economically and logistically impractical, and this creates a critical need for automated, scalable, and cost-effective monitoring solutions.

Deep learning, particularly Convolutional Neural Networks (CNNs), has emerged as a powerful tool for automating the visual inspection of solar panels. While state-of-the-art architectures like InceptionV3 and Xception can classify soiling levels with high accuracy, they come with a prohibitive computational cost. Their large model sizes and high processing demands make them fundamentally unsuitable for deployment on low-cost, resource-constrained edge devices such as a Raspberry Pi, but such devices are ideal for on-site, real-time analysis. Conversely, standard lightweight models that are designed for edge hardware often suffer from a significant drop in accuracy, forcing a difficult trade-off between predictive performance and operational efficiency.

In this paper, we aim to resolve this critical dilemma by proposing Tiny-CNN, a custom-built, ultra-lightweight CNN designed from the ground up for edge deployment. We introduce a novel two-stage optimization pipeline that transforms this compact model into a state-of-the-art soiling detector. First, we employ knowledge distillation by using a large ResNet50V2 model as a teacher to transfer its learned, complex representations to our student, Tiny-CNN, to boost its accuracy. In the second stage, we apply post-training dynamic range quantization to the distilled model, further optimizing it for on-device performance.

II. Background

This section provides a technical background of knowledge distillation and quantization, justifying their use in this study for our two-stage optimization pipeline.

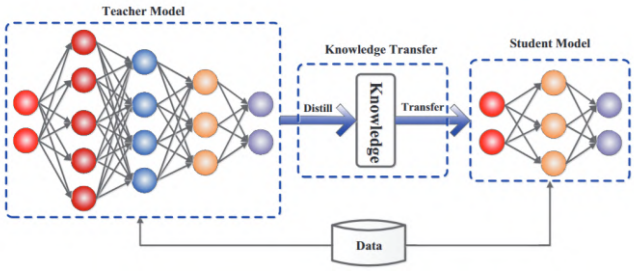


Figure 1: Teacher-student architecture for knowledge distillation [3]

A. Knowledge Distillation

The primary purpose of knowledge distillation in deep neural networks is to transfer the knowledge from more complex models known as teachers to simpler, lightweight models known as students [2] [3] [4]. This can be beneficial in overcoming hardware limitations while getting accuracies that are close to those of the more complex models.

According to [3], the training process to transfer knowledge from the teacher model to the student model, shown in Figure 1, typically involves three components. The first component is a trained, large, and complex model with high accuracy that is used as the teacher model. The second component are the soft targets from the teacher model. These soft targets are the probability distributions based on the predictions from the teacher model on the training dataset. They contain more information than just the class labels, as they also show the relative confidence of the teacher model in its predictions. The probability distributions are obtained using the softmax function:

$$q_i = \frac{exp\ (\frac{z_i}{T})}{\sum_j exp(Z_i/T)} \tag{1}$$

where q_i is the output probability for class i, z_i is the logit for class i, and T is the temperature parameter.

The third component in knowledge distillation is the student model, which is usually a lighter model with a lower accuracy than the teacher model. It is trained using both the soft targets from the teacher model and the actual labels from the dataset. In addition to the standard classification loss during training, knowledge distillation also includes a component known as distillation loss. This is a measurement of the probability distribution difference between the teacher and the student. The Kullback–Leibler (KL) function is usually used for distillation loss, and its formula is:

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} \tag{2}$$

where P and Q are probability distributions defined on the same sample space X.

The distillation loss is then calculated using the formula:

$$L_{distillation} = KL(q^T_{teacher}||q^T_{student}) \tag{3}$$

where $KL(q^T_{teacher}||q^T_{student})$ is the Kullback–Leibler divergence between the teacher's soft predictions and the student model's predictions.

According to [5], knowledge distillation makes it possible to compress the information learned by complex deep learning models during training, making it possible to deploy lighter models with this knowledge on resource-constrained devices.

B. Model Quantization

Quantization is a model compression technique that reduces the precision of a neural network's parameters and operations [6]. This precision reduction is typically from 32-bit floating point (FP32) to lower-precision formats such as 16-bit float (FP16) or 8-bit integer (INT8). This process has been shown to significantly reduce the physical size of a model, its memory bandwidth, and computational load [7]. This enables efficient inference on edge devices that have limited resources. Among the various quantization approaches, dynamic range quantization (DRQ) was chosen for this study because it offers a balance between the benefits of quantization and the ease of the quantization process. In DRQ, weights are quantized to 8-bit integers post-training, while activations are dynamically computed at runtime using floating-point values. Although it yields slightly less performance improvement than full integer quantization, DRQ still provides substantial reductions in inference latency and a four-fold reduction in model size [7]. Lower bit quantization can lower the computational requirements further, but it has been shown to cause a significant drop in model accuracy [8]. This study applied DRQ to a trained and distilled lightweight model to evaluate the performance gains from this process.

III. Related Works

Recent literature has established the superiority of deep learning over traditional computer vision for soiling classification.

Initial studies successfully demonstrated the use of large, pre-trained CNN architectures to achieve high classification accuracy. A study by Islam et al. sought to evaluate the accuracy of three state-of-the-art deep learning models in dust detection on solar panels [9]. They evaluated VGG16, VGG19, and InceptionV3 on a dataset of over 700 images, and their results showed that all three models performed well in this task. InveptionV3 had the highest accuracy at 98.3%, followed by VGG19 at 97.2%, and VGG16 had the lowest accuracy at 96.5%. Prova obtained similar results in their study, where they evaluated VGG16, InceptionV3, and a custom CNN in dust detection using a publicly available dataset from Kaggle [10]. InceptionV3 again performed the best, with an accuracy of 93.10% and an F-1 score of 94%. A similar study by Sefer et al. evaluated ten different transfer learning models in classifying images of solar panels into either clean or dirty [11]. The ten models selected ranged from lightweight models like MobileNet and EfficientNetB3 to heavier, more complex models like VGG19 and various ResNet models. Their results showed that the two best performing models were ResNet101 and ResNet50 with accuracies of 97.92% and 97.57% respectively. These studies clearly demonstrate the ability of complex models to detect soiling on solar panels accurately.

Other researchers, recognizing the need for efficiency, have explored the use of standard lightweight architectures. Haiyang et al used MobileNet for their dust detection system due to its balance of high accuracy and efficiency [12]. Abdelsattar et al. evaluated three different variants of MobileNet, namely MobileNetV1, MobileNetV2, and MobileNetV3, in detecting

dusty and clean solar panels [13]. These, the authors contend, are lightweight models with exceptional efficiency. In this study, MobileNetV1 had the highest precision at 91%, followed by MobileNetV2 at 85%, and MobileNetV3 had the lowest precision at only 62%. This study shows the potential of lightweight models in detecting the accumulation of dust particles on solar panels. Other studies that have directly compared heavier and lighter models in soiling detection have shown that the larger, more complex models tend to outperform the lighter models. This was demonstrated by Alatwi et al., where DenseNet169 had the highest accuracy compared to lighter models like MobileNet and MobileNetV2 [14]. In that study, variants of the EfficientNet lightweight models performed poorly, with all of them having less than 68% accuracy. This was also the case in the studies by Sefer and Prova, where the more complex models outperformed the simpler models [10] [11].

The need for the deployment of neural networks on devices like the Raspberry Pi has driven the development of powerful optimization techniques. Various studies have demonstrated the potential of different optimization techniques in improving model performance, including the two methods used in this study, namely knowledge distillation and quantization. In their comprehensive study, [15] evaluated the impact of quantization, pruning, and knowledge distillation on various deep learning models, namely VGGNet, ResNet50, ResNet101, and two MobileNet versions, on the CIFAR-10 and CIFAR-100 datasets. Their evaluation criteria were model accuracy, physical size, and inference latency. They showed that different quantization techniques like INT8 and FP16 resulted in a minimum of 30% reduction in model size, with their MobileNetV2 model going from 9386 KB to 2782 KB after quantization to INT8, without a decrease in model accuracy. They, however, showed that quantization came at the cost of latency across the board, with the latency of ResNet101 increasing more than tenfold when quantized to INT8, and the latency of VGGNet increasing fivefold. The study also demonstrated the effectiveness of knowledge distillation, with their evaluated models increasing in accuracy from 81.8% to 86.3% and from 81.6% to 85.4% respectively, on the CIFAR-10 dataset. Abid et al managed to reduce the size of their DenseNet121 model trained on an X-ray image dataset by 3.8 times using dynamic-range quantization [16]. They also demonstrated that DRQ can reduce model latency by between 35% and 49% in edge devices but can increase latency by up to 5880% in desktops when compared to non-quantized models. Other studies have also evaluated the different optimization techniques for edge devices with similar results [7] [17] [18].

While these optimization techniques are now well-established, their application to the specific problem of solar panel soiling detection on edge hardware has not been extensively explored. Most existing studies on PV soiling detection focus on achieving maximum accuracy with large models, whereas studies on edge AI optimization often use benchmark datasets like CIFAR-100 or COCO. Our work addresses this gap by proposing a lightweight CNN model for soiling detection on a Raspberry Pi 5 device and systematically applying both knowledge distillation and quantization to boost its accuracy and performance, thereby demonstrating a practical path toward deploying accurate and efficient PV monitoring systems in the field.

IV. Dataset Description

The foundation of this study is the Deep Solar Eye dataset [19]. This is a publicly available, large-scale collection of over 45,000 images of solar panels under real-world soiling conditions. A key advantage of this dataset is that each image is correlated with a specific power loss percentage, measured by comparing the output of the soiled panel to a clean control panel under identical environmental conditions. This provides a direct and quantifiable link between the visual appearance of a panel and its performance degradation.

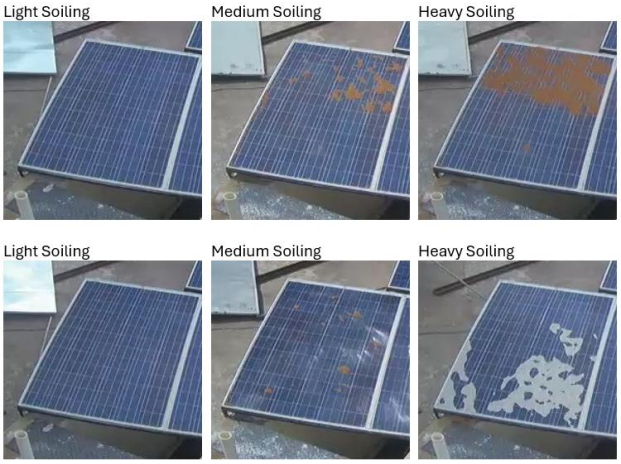


Figure 2: Sample images from the Deep Solar Eye dataset after categorization

To transform this regression-based power loss dataset into a more practical classification task suitable for actionable maintenance decisions, we partitioned the dataset into three distinct soiling classes based on the percentage power loss compared to the control panel. Sample images from each category are illustrated in Figure 2. The light soiling category is when the panels had a power loss of less than 15% when compared to the control. This category represents panels operating near their optimal level and would not warrant the cost and labour of immediate cleaning. Images in the medium soiling category are from when the sample panels had a loss of between 15% and 50% compared to the control panel. This category indicates a significant accumulation of soiling on the panel's surface that is actively hampering power output. Images in the heavy soiling category represent a critical level of soiling where the solar panel's power output is severely compromised, with more than 50% power loss, and immediate action would be required to prevent substantial energy and financial losses. This categorization allows for a tiered response strategy such as "no action needed", "cleaning recommended", and "urgent cleaning required".

V. Tiny-CNN Architecture

The main contribution of our work is Tiny-CNN, a custom-designed, lightweight CNN for soiling detection on solar panels. In contrast to leveraging complex, pre-existing architectures, our goal was to build a model from the ground

up with a singular focus on computational efficiency and a minimal memory footprint, making it an ideal candidate for resource-constrained edge devices. The architecture was implemented using TensorFlow and Keras, and all input images were standardized to a 224x224x3 resolution. The Tiny-CNN model is a sequential stack of layers, logically divided into a feature extraction base and a classification head as illustrated in Figure 3.

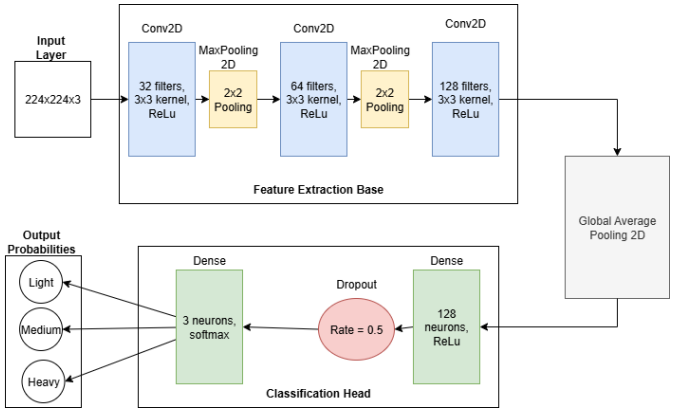


Figure 3: Tiny-CNN architecture diagram

A. Feature Extraction Base

This part of the neural network is responsible for learning hierarchical visual features from the input images. It consists of a repeating pattern of convolution and pooling layers.

Three Conv2D Layers - The network begins with a stack of three convolutional layers with filter sizes of 32, 64, and 128, respectively. The initial layers with fewer filters learn low-level features like edges, gradients, and simple textures. As the data passes through the network, subsequent layers with more filters combine these simple patterns to learn more complex and abstract features. A 3x3 kernel was used for all convolutions to capture spatial information efficiently. The ReLU activation function is used throughout to introduce non-linearity, allowing the model to learn complex relationships in the data.

Two MaxPooling2D Layers - Each of the first two convolutional layers is followed by a max-pooling layer. This operation down-samples the feature maps, reducing their spatial dimensions. This reduces the number of parameters and the computational load in subsequent layers, which is important for efficiency. It also provides a degree of translation invariance, making the model more robust to variations in the position of features within the image.

B. Classification Head

After feature extraction, this part of the network is responsible for interpreting the learned features and making a final classification.

GlobalAveragePooling2D Layer - This layer is a key design choice for ensuring our model remains lightweight. Instead of a traditional Flatten layer, which would convert the entire final feature map into a large one-dimensional vector containing thousands of parameters, global average pooling calculates the average value of each feature map. This design choice reduces the number of parameters connecting to the dense layers, thereby significantly mitigating the risk of overfitting and reducing the model's overall size.

Dense Layer with Dropout - A fully connected Dense layer with 128 neurons acts as the high-level reasoner, learning to classify the features provided by the pooling layer. This is immediately followed by a Dropout layer with a rate of 0.5 to randomly deactivate 50% of the neurons during training. This prevents the model from becoming too reliant on any single neuron or feature, which forces it to learn more robust and generalizable patterns. This is essential for preventing overfitting.

Output Layer - The final Dense layer contains three neurons, one for each of our soiling classes. It uses a softmax activation function, which converts the raw outputs into a probability distribution across the three classes, providing a clear and interpretable prediction.

VI. Two-Stage Optimization Pipeline

To enhance the performance of our Tiny-CNN model, we implemented a two-stage optimization step. This process was designed not only to prepare the model for the severe constraints of edge hardware but to also boost its predictive accuracy to a level competitive with, or even superior to, much larger models. The two stages, namely knowledge distillation and post-training quantization, illustrated in Figure 4, were applied sequentially to create the final, optimized model.

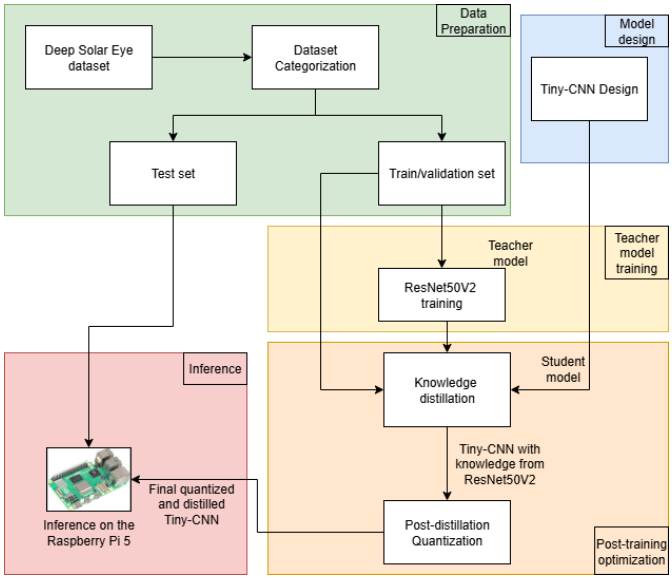


Figure 4: Schematic description of Tiny-CNN training and optimization

A. Knowledge Distillation

The first stage of our optimization process was meant to address the primary limitation of a small model, which is its capacity to learn complex data representations. To overcome this, we employed knowledge distillation to train our compact Tiny-CNN model to mimic the behaviour of a larger pretrained teacher model. For this process, we tested various high-

performing but computationally expensive models, and ResNet50v2 was selected as the teacher model due to its accuracy. Our lightweight Tiny-CNN served as the student model. During this process, the student model learned from two distinct sources of information simultaneously. It was trained using a composite loss function that combined standard cross-entropy loss which was calculated using the ground-truth hard labels from the dataset. It also learned from distillation loss, which was calculated by comparing the student's output probabilities to the soft labels produced by the teacher model. These soft labels are the full probability distributions across the three classes from the teacher's softmax layer. This output teaches the student how the teacher model generalizes and the relationships it has learned between classes. We used KL divergence to measure this loss. The core objective of the knowledge distillation stage was to transfer the rich, generalized knowledge from the deep and complex ResNet50V2 architecture into the compact and efficient Tiny-CNN.

B. Post-Training Quantization

After enhancing the model's accuracy through distillation, the second stage focused on optimizing it for on-device performance on a Raspberry Pi device. We applied post-training dynamic range quantization to the Tiny-CNN model. This process converts the model's FP32 weights to a more efficient INT8 representation. Unlike full integer quantization, which requires a calibration dataset and full INT8 arithmetic, DRQ is simpler to apply and does not necessitate retraining or the creation of representative datasets. We applied DRQ using the tensorflow.lite.Optimize.DEFAULT library, and this converted the Tiny-CNN's weight parameters from FP32 to INT8 to reduce memory usage and improve latency, while activations remained in floating-point format. Due to its efficiency and ease of implementation, symmetric quantization was used. During inference on the Raspberry Pi's CPU, the TensorFlow Lite interpreter dequantizes these INT8 weights back to FP32 values before performing the multiply–accumulate operations. This means that although the model benefits from the compact storage and faster memory access of quantized weights, the arithmetic is still carried out in floating point to preserve numerical accuracy. The activations and intermediate tensors are maintained in FP32 precision throughout the process. This hybrid inference mode offers an effective trade-off between efficiency and model accuracy, which is ideally suited to our chosen Raspberry Pi 5 edge device.

VII. RESULTS AND ANALYSIS

This section presents the empirical results of our study, quantitatively demonstrating the effectiveness of our proposed optimization pipeline. We first establish the performance benchmarks of existing architectures before detailing the step-by-step enhancement of our custom Tiny-CNN model. All models were trained on a computer with an Intel Core i5 12450HX processor, an Nvidia GeForce RTX 4060 GPU, and 32GB of RAM. All performance metrics were recorded on a Raspberry Pi 5 device with 4GB of RAM.

A. Performance Evaluation of Benchmark Architectures

To contextualize our findings, we first evaluated several standard deep learning models. As shown in Table 1, the results confirm the well-established trade-off between model complexity and operational efficiency. Heavyweight models like InceptionV3 and ResNet50v2 achieved high classification accuracies of 91.93% and 91.62% respectively, but were operationally impractical, with the best performing model InceptionV3 requiring over 343 ms for a single inference. Standard lightweight alternatives like MobileNet offered a better balance, achieving a strong accuracy of 91.81% with a much faster inference time of 36.74 ms, yet its model size remained substantial at over 12 MB. These benchmarks largely illustrate the challenge that this paper sought to solve, of trying to create a model that is more accurate than the heavyweight models while being significantly faster and smaller than the lightweight ones.

Table 1: Performance metrics of all the evaluated models on a Raspberry Pi 5

Model	**Accuracy**	**F1 Score**	**Model size (KB)**	**Average inference time per image (ms)**	**Average RAM per image (MB)**
Heavyweight Models					
DenseNet121	90.74	88.25	27248	187.31	225.2
DenseNet169	90.92	87.97	48885	225.24	272.53
DenseNet201	91.13	87.97	70814	294.62	333.81
InceptionV3	91.93	89.39	85107	343.09	332.86
Xception	91.52	89.04	81248	530.84	359.36
VGG16	87.98	84.93	57495	927.3	295.06
ResNet101V2	89.67	87.28	91913	462.8	519.07
ResNet50V2	91.62	88.8	166177	228.85	353.92
Lightweight Models					
MobileNet	91.81	89.11	12512	36.74	179.88
MobileNetV2	86.4	80.63	8666	22.16	173.67
Our Proposed work					
Tiny-CNN (Baseline)	88.21	83.27	435	29.07	184.66
Tiny-CNN + Distillation	92.26	89.9	435	29.1	168.17
Tiny-CNN + Distillation + Quantization	**92.27**	**89.92**	**119**	**18.18**	**166.46**

B. *Tiny-CNN Performance Evaluation*

Our custom Tiny-CNN was designed for efficiency, and its performance improvements through our two-stage optimization pipeline showcases the power of this approach. In its initial, unoptimized state, our baseline Tiny-CNN was already highly efficient. It achieved an inference time of 29.07 ms and had a minimal footprint of just 435 KB. However, its accuracy of 88.21% and F1-score of 83.27% lag behind the larger benchmark models, making it an inadequate solution on its own and justifying the need for further enhancement.

The application of knowledge distillation from the ResNet50v2 teacher model marked a pivotal transformation. As detailed in Table 1, the accuracy of the distilled Tiny-CNN improved from 88.21% to 92.26%. This is the first key finding of our work, that through knowledge distillation, our simple, custom architecture did not just match, but surpassed the accuracy of its complex teacher. This was achieved with no change to the model's size or its inference time, immediately creating a model that was both more accurate and over 7 times faster than its teacher.

The final optimization stage was post-training dynamic range quantization, and this yielded our most significant and surprising results. The impact was threefold. Firstly, the model's file size shrank from 435 KB to a mere 119 KB, a 72% reduction that makes it trivial to store and deploy. The average inference time also dropped by 37%, from 29.1 ms to just 18.18 ms, making it twice as fast as the MobileNet benchmark and the fastest model in our study. Most notably, quantization did not degrade performance but instead provided a slight increase in accuracy to 92.27% and an F1-score of 89.92%. This is a critical finding that challenges the common expectation of a performance trade-off with post-training quantization. We hypothesize that this improvement may stem from a regularization effect, where the reduced precision of the weights helps to smooth the decision boundary and prevent overfitting, leading to better generalization on the test set.

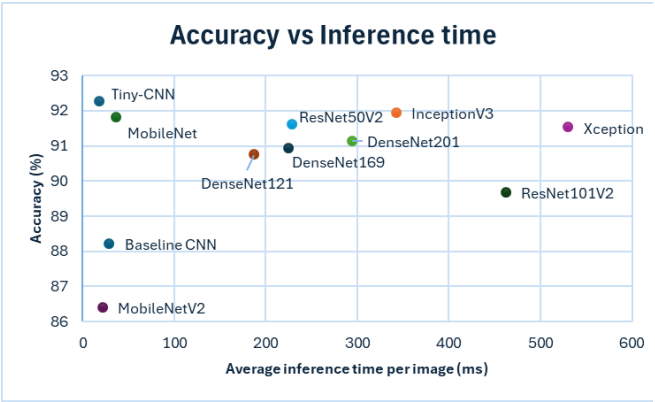


Figure 5: Plot of accuracy vs inference time per image for all models evaluated

The final, optimized Tiny-CNN is a comprehensively superior solution for this task compared to the other models tested. It is simultaneously the most accurate, fastest, and smallest model evaluated. To visually represent this, Figure 5 shows a plot of all benchmarked models based on their accuracy versus inference speed. The heavyweight models occupy the slow-but-accurate top-right quadrant, while our final model sits in the ideal top-left quadrant, clearly demonstrating its state-of-the-art performance and efficiency.

The success of our two-stage optimization pipeline provides a clear blueprint for developing highly effective edge AI solutions. By first using a powerful teacher to distil knowledge to a simple student architecture, and then applying aggressive optimization like quantization, it is possible to create field-ready models that break the traditional constraints of edge computing.

VIII. LIMITATIONS AND FUTURE WORK

The proposed two-stage optimization pipeline still needs to undergo further testing to evaluate whether the results obtained in this study are transferable across different datasets and fields of study beyond soiling detection. Other optimization techniques, such as pruning, will also be explored to assess their impact on accuracy and performance on edge devices.

IX. CONCLUSION

In this work, we successfully addressed the critical trade-off between accuracy and efficiency for automated solar panel soiling detection on resource-constrained edge devices. We presented Tiny-CNN, a custom lightweight neural network enhanced by a novel two-stage optimization pipeline of knowledge distillation and post-training quantization. Our results demonstrate the success of this approach. The final optimized model achieves a state-of-the-art accuracy of 92.27%, outperforming the best benchmark model and its large ResNet50V2 teacher model, while being orders of magnitude more efficient. It performs inference in just 18.18 ms on a Raspberry Pi 5 and has a final footprint of only 119 KB. Crucially, we show that quantization can not only improve efficiency but also slightly enhance accuracy, challenging common assumptions.

ACKNOWLEDGMENT

This work was carried out in the Distributed Multimedia CoE at Rhodes University and at the Copperbelt University in Zambia with financial support from the European Union through the Grow Green Africa consortium and from Telkom. The authors acknowledge that the opinions, findings and conclusions or recommendations expressed here are those of the authors and that none of the above-mentioned sponsors accept any liability whatsoever in this regard.

REFERENCES

[1] S. Isaacs, O. Kalashnikova, M. J. G. b, A. v. Donkelaar, M. S. Hammer, H. Lee and D. Wood, "Dust soiling effects on decentralized solar in West Africa," *Applied Energy*, vol. 340, no. 15, p. 120993, 2023.

[2] G. Hinton, O. Vinyals and J. Dean, "Distilling the Knowledge in a Neural Network," Google, Mountain View, 2015.

[3] J. Gou, B. Yu, S. J. Maybank and D. Tao, "Knowledge distillation: A survey," *International journal of computer vision,* vol. 129, no. 6, pp. 1789-1819, 2021.

[4] A. G. Belinga, C. Stéphane, T. Koumetio, M. E. Haziti and M. E. Hassouni, "Knowledge Distillation in Image Classification: The Impact of Datasets," *Computers,* vol. 13, no. 8, p. 184, 2024.

[5] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari and K. He, "Data distillation: Towards omni-supervised learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[6] J. Wu, C. Leng, Y. Wang, Q. Hu and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[7] B. Rokh, A. Azarpeyvand and A. Khanteymoori, "A Comprehensive Survey on Model Quantization for Deep Neural Networks in Image Classification," *ACM Transactions on Intelligent Systems and Technology,* vol. 14, no. 6, pp. 1-50, 2023.

[8] B. Li, S. Qu and Y. Wang, "An automated quantization framework for high-utilization rram-based pim," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 41, no. 3, pp. 583-596, 2021.

[9] S. Islam, M. T. Zaman, H. Anonto, T. Banik, I. Hossain and A. Shufian, "Enhancing Solar Panel Efficiency through Advanced Dust Detection: A Comparative Study of VGG16, VGG19, and Inception V3 Convolutional Neural Networks," in *2025 2nd International Conference on Advanced Innovations in Smart Cities (ICAISC)*, Jeddah, 2025.

[10] N. N. I. Prova, "Improved Solar Panel Efficiency through Dust Detection Using the InceptionV3 Transfer Learning Model," in *2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Kirtipur, 2024.

[11] T. Sefer and M. Kaya, "Detection of Dust on Solar Panels with Deep Learning," *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi,* vol. 27, no. 4, pp. 1451-1464, 2024.

[12] H. He, C. Zhou, P. Yu and X. Lu, "Research on a Photovoltaic Panel Dust Detection System Based on Improved Mobilenet Algorithm," in *2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, 2024.

[13] M. Abdelsattar, A. AbdelMoety and A. Emad-Eldeen, "Detecting Dusty and Clean Photovoltaic Surfaces Using MobileNet Variants for Image Classification," *SVU-International Journal of Engineering Sciences and Applications,* vol. 6, no. 1, pp. 9-18, 2025.

[14] A. M. Alatwi, H. Albalawi, A. Wadood, H. Anwar and H. El-Hageen, "Deep Learning-Based Dust Detection on Solar Panels: A Low-Cost Sustainable Solution for Increased Solar Power Generation," *Sustainability,* vol. 16, no. 19, p. 8664, 2024.

[15] H. Lee, N. Lee and S. Lee, "A Method of Deep Learning Model Optimization for Image Classification on Edge Device," *Sensors,* vol. 22, no. 19, p. 7344, 2022.

[16] A. Abid, P. Sinha, A. Harpale, J. Gichoya and S. Purkayastha, "Optimizing medical image classification models for edge devices," in *International Symposium on Distributed Computing and Artificial Intelligence*, 2021.

[17] W. Chen, H. Qiu, J. Zhuang, C. Zhang, Y. Hu, Q. Lu, T. Wang, Y. Shi, M. Huang and X. Xu, "Quantization of deep neural networks for accurate edge computing," *ACM Journal on Emerging Technologies in Computing Systems (JETC),* vol. 17, no. 4, pp. 1-11, 2021.

[18] S. Naveen and M. R. Kounte, "Optimized convolutional neural network at the IoT edge for image detection using pruning and quantization," *Multimedia Tools and Applications,* vol. 84, no. 9, pp. 5435-5455, 2025.

[19] S. Mehta, A. P. Azad, S. A. Chemmengath, V. Raykar and S. Kalyanaraman, "DeepSolarEye: Power Loss Prediction and Weakly Supervised Soiling Localization via Fully Convolutional Networks for Solar Panels," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Tahoe, 2018.

Wireless Communications and Next-Generation Technologies

Outage Performance Analysis of Green Energy IoT Network

1st Neo Selematsela
School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg, South Africa
1262347@students.wits.ac.za

2nd Chabalala Chabalala
School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg, South Africa
chabalala.chabalala@wits.ac.za

3rd Fambirai Takawira
School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg, South Africa
fambirai.takawira@wits.ac.za

***Abstract*—This paper presents performance analysis of green energy IoT network, where the base-station is powered by natural energy sources, in this case solar energy. To have a complete green energy system, the IoT devices also harvest energy from downlink Radio Frequency (RF) transmissions of the base-station to perform their uplink transmissions. For performance analysis, this paper treats the energy battery storage at the base station and the one at the IoT devices as tandem queues, and then examines the energy flows analysis between the two tandem energy queues using tandem queue decomposition method. Using the derived energy flow analysis model, the study then derives the analytical expression for the considered IoT node uplink energy-aware outage probability. The analysis carried-out in this paper forms the basis for informed deployments of green energy IoT networks and optimization considerations. The analytical expressions derived are verified by comparing their simulation and analytical results using Matlab.**

***Index Terms*—Energy Harvesting, IoT, Markovian Analysis, Outage Probability, Tandem Queues.**

I. Introduction

Green Energy IoT wireless networks is an evolving area that combines sustainable energy solutions with smart connected technologies. The internet of things (IoT) has a significant economic and environmental impact owing to the billions or trillions of interconnected devices that use various types of sensors to communicate through the internet. It is well-recognized that each sensor requires a small amount of energy to function; but, with billions of sensors, energy consumption can be significant. Therefore, it is crucial to focus on developing energy-efficient IoT technology and sustainable solutions [1]. Green IoT envisions the concept of reducing the energy consumption of IoT devices and making the environment safe [2].

Recent technological advances have led to an increase in the carbon footprint. Energy efficiency in the Internet of Things (IoT) has been attracting a lot of attention from researchers and designers over the last couple of years, paving the way for an emerging area called green IoT [3]. To improve energy efficiency in wireless networks and reduce carbon foot-print, base stations and IoT devices are deployed with the capability to harvest energy from ambient sources. The considered system model in this paper assumes the base station harvest energy from solar and IoT devices with a capability to harvest energy from the RF downlink transmissions from the base station.

Multiple research works have modelled and analysed performance for energy harvesting IoT wireless networks. The study in [4], proposes a model that amalgamates different wave propagation mechanisms for signal strength prediction in the propagation of RF-waves, this helps in improving the modelling accuracy for RF-waves in RF energy harvesting. Authors in [5], present an overview of enabling technologies for efficient WEH, analyzing the lifetime of WEH-enabled IoT devices. The study in [6], highlights future design challenges of IoT energy harvesters that must be addressed to continuously and reliably deliver energy. These works discuss design approaches for efficient energy harvesting process in IoT networks different from the work done in this paper, where we focus on performance analysis of a network model that has both the base station and IoT devices harvesting energy from ambient sources.

The studies in [7]–[11] analyze the performance of energy harvesting networks, including outage probability analysis. However, their system models do not fully utilize green energy, as either the base station or the IoT devices harvest energy from ambient sources (not both simultaneously) and therefore do not represent a fully off-grid green energy system, as proposed in this paper. The studies in [12]–[15] provide an analysis of tandem queues either from a queuing theory perspective or through their application to wireless network models. In contrast, this paper applies tandem queue analysis concepts to model energy flow between a solar energy harvesting base station and RF energy harvesting IoT devices. This energy flow model is then used to develop an efficient power management strategy for the IoT devices, enabling transmit power control and outage probability analysis. To the best of the authors' knowledge, the work presented in this paper has not been previously published.

Identify applicable funding agency here. If none, delete this.

A. Contributions

The contributions of this work can be summarized as follows:

- Develop a comprehensive energy flow analysis between a green-energy-powered base station and RF energy-harvesting (EH) IoT device. To capture the dynamics of energy harvesting, storage, and consumption, the system is modelled as a tandem queue, where the first queue represents the base station's renewable energy queue and the second queue represents the energy queue of the IoT device. Using the tandem queue decomposition method, system steady-state probabilities for both queues is derived, this provides insight into the long-term availability and distribution of energy across the network.
- Closed-form expression for the outage probability of the considered RF energy-harvesting (EH) IoT device is derived, where outage depends jointly on the uplink transmission channel conditions and the availability of energy in the IoT device energy queue. Building on this formulation, outage performance analysis is conducted by systematically varying key network parameters. This allows for quantifying sensitivity of outage probability to both energy and uplink channel dynamics, providing insights into the fundamental trade-offs between energy availability and communication reliability in EH-enabled IoT networks.

II. System Model

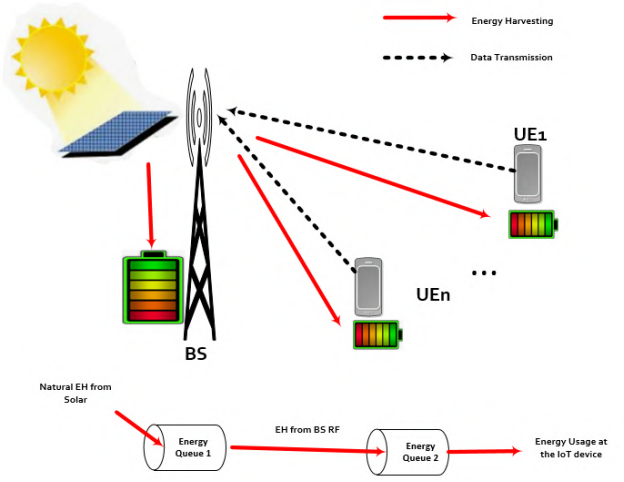


Fig. 1. Green Energy IoT Network

In this study, we consider green energy IoT network nodes where the base-station (BS) harvest energy from ambient sources in this case solar powered and the IoT nodes (UEs) harvest their energy from the radio frequency (RF) transmissions from the BS as depicted in Fig. 1. The considered EH IoT network is made up of one Base-Station (BS) and multiple IoT devices connected to the BS. For simplicity, our analysis will be done with one IoT node (UE_n) in the considered IoT network, where n is the index of each IoT device. We assume a time-slotted system where the UEs are allocated time-slots to use for their transmissions. In a given transmission time-slotted frame, the UEs use the allocated time-slot for their transmissions and then perform energy harvesting from the BS downlink RF for the remainder of the time-slots. We further assume a quasi-static network scenario, where the state of the UEs remain unchanged for the duration of each time frame. The frames are indexed as follows: t = {0,1,2,3,..}.

TABLE I
TABLE OF SYMBOLS

Symbol	Description
$\lambda_{G_{bs}^{UE_n}}$	downlink channel rate between BS and UE_n for energy harvesting.
$G_{bs}^{UE_n}$	downlink channel gain between SBS and SU_n for energy harvesting.
$\lambda_{H_n^{bs}}$	uplink channel rate between UE_n and BS.
H_n^{bs}	uplink channel gain between UE_n and BS.
P_n^{UE}	uplink transmit power from UE_n to BS.
P_{bs}^{T}	downlink transmit power from BS to the UEs for energy harvesting.
R^T	is the target rate for the UE_n.
γ_n	is the target signal to noise and interference ratio (SINR) for each UE_n on the uplink at the BS.
$E_n^H(t)$	is the energy harvested by the UE_n during the frame index t.
$E_{bs}^H(t)$	is the energy harvested by the BS from solar in frame index t.
$E_n^C(t)$	is the energy consumed by the UE_n during the frame index t.
$E_{bs}^C(t)$	is the energy consumed by the BS on downlink transmission.
W	is the allocated network bandwidth.
η	energy harvesting efficiency.
σ^2	Additive White Gaussian Noise (AWGN).
Δ_p	battery power discretization steps.
T_f	frame duration.
t_{ts}	transmission time-slot duration.
$(T_f - t_{ts})$	energy harvesting duration.
UE_n	IoT user in the considered network.
$b_n(t)$	available battery energy of UE_n at frame index t.
$b_{bs}(t)$	available battery energy of BS at frame index t.
BS	Base station.

A. Base Station Energy Dynamics

The BS harvests energy from the solar panels and transmit with a fixed power, $P_{bs}^{T_x}$. To model energy harvesting process by the BS we take into account the stochastic change in solar power which is affected by the different seasons in a year, weather, and other effects. Therefore, the proposed solar energy model in this work is based models described in [11], [16], [17], which capture multiple parameters that affect the amount of solar irradiance measured in watts per unit area (watts/meters^2). The parameters include, direct beam radiation from the sun, the diffuse radiation that gets scattered out by different objects on the environment, and the radiation that gets reflected off the ground into the atmosphere. Average solar radiation models in terms of the day, , is expressed as follows,

$$R(\tau) = \frac{h_{peak}}{2}\left[1 + cos\left(\frac{\pi}{6}(\tau - 12)\right)\right] + A_r \times N(\tau, 1), \tag{1}$$

where:

- τ is assumed to be $6 \leq \tau \leq 18$ and otherwise 0.
- h_{peak} is the peak value of power harvesting dependent on season or weather.
- A_r is the amplitude of the unexpected changes.
- $N(\tau, 1)$ is the Guassian distribution with mean τ and unit variance.

$A_r \times N(\tau, 1)$ is assumed to capture the random changes that can occur from cloud, rain, or other effects.

Following the similar approach as in [17], the harvested solar power by the BS at the certain time of the day $\tau = \{1, 2, 3, ..., 24\}$ can be predicted as follows,

$$P_{sol}^{bs}(t) = (1 - L(M)) \times e_{panel} \times e_{el} \times \theta \times A \times R(\tau), \tag{2}$$

where:

- $L(M)$ is the energy loss due to inability to store solar power by the BS when the battery temeratures are too high or too low, e.g., for lithium batteries it is temperatures $< 0^0C$ or those $> 45^0C$, where M is a constant used to represent the saturation output power for the EH circuit.
- A is the solar panel size.
- θ angular loss to get the received radiation at the specific time τ.
- e_{panel} is the solar panel efficiency losses.
- e_{el} is the losses during electrical conversion.

Battery bank at the BS is utilized to store energy generated by the solar panels to regulate system voltage and supply energy to the BS [11]. The lithium battery is considered for its features like high-energy density, long endurance time, and easy maintenance. The energy state of the BS battery bank at some time frame t can be expressed as follows,

$$b_{bs}(t) = min\{max\{b_{bs}(t-1) + E_{bs}^{H}(t-1) - E_{bs}^{C}(t-1), 0\}, B_{bs}^{max}\}. \tag{3}$$

Using Equations (2) and (3) as the reference, BS harvested energy expressions,$E_{bs}^{H}(t)$, at the tth frame index are expressed as follows, respectively.

$$\begin{aligned} E_{bs}^{H}(t) &= P_{sol}^{bs}(t) \times T_F \\ &= \left[(1 - L(M)) \times e_{panel} \times e_{el} \times \theta \times A \times R(\tau)\right] \\ &\times T_F. \end{aligned} \tag{4}$$

The cumulative density function (CDF) for BS energy harvesting, $E_{bs}^{H}(t)$, is derived as follows,

$$\begin{aligned} F_{E_{bs}^{H}}(x) &= Pr\left(E_{bs}^{H} < x\right) \\ &= Pr\left(P_{sol}^{bs} \times T_F < x\right) \\ &= Pr\Bigg(\Bigg((1 - L(M)) \times e_{panel} \times e_{el} \times \theta \\ &\times A \times R(\tau)\Bigg) \times T_F < x\Bigg) \\ &= Pr\Bigg(\Bigg(\frac{h_{peak}}{2}\left[1 + cos\left(\frac{\pi}{6}(\tau - 12)\right)\right] \\ &+ A_r \times N(\tau, 1)\Bigg) \\ &\times \Bigg((1 - L(M)) \times e_{panel} \times e_{el} \times \theta \times A \times T_F\Bigg) < x\Bigg) \\ &= Pr\Bigg(\Phi_1 + \Phi_2 N(\tau, 1) < x\Bigg) \\ &= Pr\left(N(\tau, 1) < \frac{x - \Phi_1}{\Phi_2}\right), \end{aligned} \tag{5}$$

where,

- $\Phi_1 = \frac{h_{peak}}{2}\left[1 + cos\left(\frac{\pi}{6}(\tau - 12)\right)\right] \times \Bigg((1 - L(M)) \times e_{panel} \times e_{el} \times \theta \times A \times T_F\Bigg)$.
- $\Phi_2 = A_r \times \Bigg((1 - L(M)) \times e_{panel} \times e_{el} \times \theta \times A \times T_F\Bigg)$.

Similar to [18], analytical expression for the CDF of E_{bs}^{H} is as follows,

$$\begin{aligned} F_{E_{bs}^{H}}(x) &= Pr\left(N(\tau, 1) < \frac{x - \Phi_1}{\Phi_2}\right) \\ &= \frac{1}{2}\left(1 + erf\left(\frac{x - \Phi_1 - \Phi_2\tau}{\Phi_2\sqrt{2}}\right)\right), \end{aligned} \tag{6}$$

where erf is the error function. Similarly, probability density function (PDF) is expressed as follows, [19],

$$\begin{aligned} f_{E_{bs}^{H}}(x) &= \frac{d}{dx} Pr\left(N(\tau, 1) < \frac{x - \Phi_1}{\Phi_2}\right) \\ &= \frac{d}{dx} F_{N(\tau,1)}\left(\frac{x - \Phi_1}{\Phi_2}\right) \\ &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x - \Phi_1 - \Phi_2\tau}{\Phi_2}\right)^2} \end{aligned} \tag{7}$$

BS energy consumption is expressed as follows, where BS donwlink transmit power, $P_{bs}^{T_x}$, is fixed.

$$E_{bs}^{C}(t) = P_{bs}^{T_x} \times t_{ts} \tag{8}$$

B. UE Energy Dynamics

The UEs are equipped with two Radio Frequency (RF) chains, one for transmission on the uplink frequency, while the second one switches between data reception and energy harvesting on the downlink frequency. The UEs receive data transmissions on the allocated downlink time-slot and switch to energy harvesting for the remainder of the frame duration. UEs transmit power is selected as follows,

$$P_n^{UE}(t) = min\left(\frac{b_n(t) - b_{th}}{t_{ts}}, P_n^R(t)\right), \tag{9}$$

where P_n^R is the target rate power and is expressed as follows using Shannon capacity theorem [20],

$$P_n^R(t) = \frac{\left(2^{\frac{R^T}{W}} - 1\right)\sigma^2}{H_n^{bs}(t)}. \tag{10}$$

In the uplink scenario, each UE transmits its individual signal x_n towards the BS. The generalized received signal expression for each UE at the BS is defined as follows [21],

$$y_n(t) = \sqrt{P_n^{UE}(t)H_n^{bs}(t)}x_n(t) + w(t). \tag{11}$$

Using Equation (11), the instantaneous signal to noise and interference ratio (SINR) expression for each UE_n is defined as follows,

$$\gamma_n(t) = \frac{P_n^{UE}(t)H_n^{bs}(t)}{\sigma^2}. \tag{12}$$

The battery energy evolution for the UE can be presented by the below equation where frame indexing is at the beginning of the frame, similar as in [22],

$$b_n(t) = min\{max\{b_n(t-1) + E_n^H(t-1|i) - E_n^C(t-1), 0\}, B_n^{max}\}. \tag{13}$$

Using Equation(13) as reference, UE energy harvested, $E_n^H(t)$, at the tth frame index are expressed as follows,

$$\begin{aligned} E_n^H(t|i) &= \left(\eta\left(T_f - t_{ts}\right)P_{bs}^{T_x}G_{bs}^{UE_n}(t)\right) \\ &= \left(\eta\left(T_f - t_{ts}\right)\left(\frac{n_i\Delta_p}{t_{ts}}\right)G_{bs}^{UE_n}(t)\right). \end{aligned} \tag{14}$$

where $P_{bs}^{T_x} = \frac{n_i\Delta_p}{t_{ts}}$ is the BS transmit power which is fixed at energy units n_i. Thus, the BS consumes fixed number of i discritized energy units for its transmissions.

Since the UE only harvests when the BS is transmitting, the UE energy harvesting depends on BS energy level not being less than n_i energy units, which is the energy units required to support the fixed BS transmit power, $P_{bs}^{T_x}$. Thus, the CDF for UE energy harvesting is derived as follows,

$$\begin{aligned} F_{E_n^H(t)}(x)|_{i \geq n_i + b_{th}^{bs}} &= Pr\left(E_n^H(t) < x\right) \\ &= Pr\left(\eta\left(T_f - t_{ts}\right)\left(\frac{n_i\Delta_p}{t_{ts}}\right)G_{bs}^{UE_n}(t) < x\right) \\ &= 1 - e^{-\left(\frac{\lambda_{G_{bs}^{UE_n}(t)}{}^{xt_{ts}}}{\eta\left(T_f - t_{ts}\right)n_i\Delta_p}\right)}. \end{aligned} \tag{15}$$

Unconditioning on the values i would take, we get the below CDF expression.

$$\begin{aligned} F_{E_n^H(t)}(x) &= Pr\left(E_n^H(t) < x\right) \times Pr\left(i \geq n_i + b_{th}^{bs}\right) \\ &= \sum_{i=n_i+b_{th}^{bs}}^{\frac{B_{bs}^{max}}{\Delta_p}} \left(1 - e^{-\left(\frac{\lambda_{G_{bs}^{UE_n}(t)}{}^{xt_{ts}}}{\eta\left(T_f - t_{ts}\right)n_i\Delta_p}\right)}\right)\Pi_{bs}(i) \end{aligned} \tag{16}$$

Similarly, UE energy consumed at the tth frame index are expressed as follows,

$$\begin{aligned} E_n^C(t) &= P_n^{UE}(t) \times t_{ts} \\ &= \left(min\left(\frac{b_n(t) - b_{th}^{UE}}{t_{ts}}, P_n^R(t)\right)\right) \times t_{ts}. \\ &= \left(min\left(\frac{k\Delta_p - b_{th}^{UE}}{t_{ts}}, \frac{\left(2^{\frac{R^T}{W}} - 1\right)\sigma^2}{H_n^{bs}(t)}\right)\right) \times t_{ts}. \end{aligned} \tag{17}$$

CDf for UE energy consumed is derived as follows from Equation (17) above,

$$\begin{aligned} F_{E_n^C(t)}(x) = Pr\Bigg(&\left(min\left(\frac{k\Delta_p - b_{th}^{UE}}{t_{ts}}, \frac{\left(2^{\frac{R^T}{W}} - 1\right)\sigma^2}{H_n^{bs}(t)}\right)\right) \\ &\times t_{ts} < x\Bigg) \\ = 1 - Pr\Bigg[&Pr\left(k \geq \frac{x + b_{th}^{UE}}{\Delta_p}\right) \\ &\times Pr\left(H_n^{bs}(t) \geq \frac{\left(2^{\frac{R^T}{W}} - 1\right)\sigma^2 t_{ts}}{x}\right)\Bigg] \\ = 1 - \Bigg[&\sum_{k=\frac{x+b_{th}^{UE}}{\Delta_p}}^{B_n^{max}} \Pi_n(k) \\ &\times \left(1 - e^{-\left(\frac{\lambda_{H_n^{bs}(t)}\left(2^{\frac{R^T}{W}} - 1\right)\sigma^2 t_{ts}}{x}\right)}\right)\Bigg] \end{aligned} \tag{18}$$

III. System Energy Flow Analysis

Energy flow in the considered system moves from BS battery buffer to the UE battery through RF transmission, in which the energy in to the BS battery is harvested from the sun using solar panels. We model the system energy flow as a tandem queue of two queues, Q_{bs} as energy queue 1 and Q_{bs} as energy queue 2 shown in Fig.1.

Queue Q_{bs} has external energy arrivals dependent on energy harvesting process from the solar panels which follows a Gaussian distribution with mean τ and unit variance as shown in Equation (4). Since BS transmit power is fixed as shown in Equation (8), service time is deterministic. BS battery capacity is considered to have a finite size of B_{bs}^{max}, including the energy chunks in service if any. Therefore, Q_{bs} is represented as follows using classic Kendall's notation,

$$G/D/1/B_{bs}^{max} \tag{19}$$

Queue Q_n has arrivals that are dependent on departures from Q_{bs} because the UE harvests energy from the downlink RF transmission of the BS. Thus, Q_n has general distribution for inter-arrivals, with the distribution dependent on UE RF energy harvesting process as shown Equation (16). Q_n service distribution follows a general distribution that is dependent on the UE uplink transmit power selection as shown in Equation (17). Maximum capacity for Q_n is set at B_n^{max}. Q_n queueing model can be expressed as follows using Kendall's notation,

$$G/G/1/B_n^{max} \tag{20}$$

The considered queuing system model is represented as a two-dimensional discrite time Markov chain as follows

$$\{q_{bs}(t), q_n(t), t \geq 0\}, \tag{21}$$

where $q_{bs}(t)$ and $q_n(t)$ represents the number of energy units at time t in Q_{bs} and Q_n, respectively.

The above two-dimensional system Markov chain has the state space represented as follows,

$$\begin{aligned} S = \{&(0,0),(0,1),...,(0,\frac{B_n^{max}}{\Delta_p}); \\ &(1,0),(1,1),...,(1,\frac{B_n^{max}}{\Delta_p}); \\ &... \\ &(\frac{B_{bs}^{max}}{\Delta_p},0),(\frac{B_{bs}^{max}}{\Delta_p},1),...,(\frac{B_{bs}^{max}}{\Delta_p},\frac{B_n^{max}}{\Delta_p})\} \end{aligned} \tag{22}$$

Let $Q_{bs,n}((i,k),(j,l))$ represent the joint transition probability from state (i,k) to (j,l) where the BS energy queue evolves from state i to j and the UE energy queue evolves from k to l. This joint transition probability is characterized by the following state conditions considering the transition events that the BS and the UE can take.

Condition (1): $i \leq \frac{b_{th}^{bs}}{\Delta_p}$ in which the BS does not transmit, thus $E_{bs}^C = 0$ and the UE does not harvest, $E_n^H = 0$, thus the joint state transition probability is represented as follows using Q_{bs} and Q_n energy evolution Equations (3) and (13), respectively.

$$\begin{aligned} Q_{bs,n}((i,k),(j,l)) &= Pr\left(j = i + \frac{E_{bs}^H}{\Delta_p}\right) \\ &\times Pr\left(l = k - \frac{E_n^C}{\Delta_p}\right) \\ &= Pr\left(E_{bs}^H = (j-i)\,\Delta_p\right) \\ &\times Pr\left(E_n^C = (l-k)\,\Delta_p\right) \\ &= \Bigg(F_{E_{bs}^H}\left((j-i+1)\,\Delta_p\right) \\ &- F_{E_{bs}^H}\left((j-i)\,\Delta_p\right)\Bigg) \\ &\times \Bigg(F_{E_n^C}\left((l-k+1)\,\Delta_p\right) \\ &- F_{E_n^C}\left((l-k)\,\Delta_p\right)\Bigg). \end{aligned} \tag{23}$$

Condition (2): $i > \frac{b_{th}^{bs}}{\Delta_p}$ in which the BS is transmitting, thus $E_{bs}^C > 0$ and the UE is harvesting, $E_n^H > 0$. Similarly, the joint state transition probability is represented as follows,

$$\begin{aligned} Q_{bs,n}((i,k),(j,l)) &= Pr\left(j = i + \frac{E_{bs}^H}{\Delta_p} - \frac{E_{bs}^C}{\Delta_p}\right) \\ &\times Pr\left(l = k + \frac{E_n^H(i)}{\Delta_p} - \frac{E_n^C}{\Delta_p}\right) \\ &= Pr\left(\frac{E_{bs}^H}{\Delta_p} = \left((j-i+\frac{E_{bs}^C}{\Delta_p}\right)\right) \\ &\times Pr\left(\frac{E_n^H(i)}{\Delta_p} = \left(l-k+\frac{E_n^C}{\Delta_p}\right)\right), \\ &= Pr\left(E_{bs}^H = \left(j-i+\frac{E_{bs}^C}{\Delta_p}\right)\Delta_p\right) \\ &\times Pr\left(E_n^H(i) = \left(l-k+\frac{E_n^C}{\Delta_p}\right)\Delta_p\right). \end{aligned} \tag{24}$$

Using Equations (6) and (15), then let $E_n^C = u\Delta_p$ and $E_{bs}^C = v\Delta_p$. The closed-form expression for the above joint transition probability is as follows.

$$
\begin{aligned}
Q_{bs,n}((i,k),(j,l)) &= Pr\left(E_{bs}^{H} = (j - i + \frac{E_{bs}^{C}}{\Delta_p})\Delta_p\right) \\
&\times Pr\left(E_{n}^{H}(i) = (l - k + \frac{E_{n}^{C}}{\Delta_p})\Delta_p\right) \\
&= \Bigg(F_{E_{bs}^{H}}\left((j - i + v + 1)\Delta_p\right) \\
&- F_{E_{bs}^{H}}\left((j - i + v)\Delta_p\right)\Bigg) \\
&\times \Bigg(\sum_{u=\frac{b_{th}^{UE}}{\Delta_p}}^{\frac{B_n^{max}}{\Delta_p}} \Bigg[F_{E_{n}^{H}}\left(l - k + u + 1)\Delta_p\right) \\
&- F_{E_{n}^{H}}\left(l - k + u)\Delta_p\right)\Bigg]\Pi_n(u)\Bigg)
\end{aligned} \tag{25}
$$

In steady-state, the joint probability of the system being in any state (j, l) at time t is represented as follows using Markovian analysis,

$$
\Pi_{bs,n}(j,l) = \sum_{i=0}^{\frac{B_{bs}^{max}}{\Delta_p}} \sum_{k=0}^{\frac{B_n^{max}}{\Delta_p}} Q_{bs,n}((i,k),(j,l)) \times \Pi_{bs,n}(i,k), \tag{26}
$$

where

$$
\sum_{j=0}^{\frac{B_{bs}^{max}}{\Delta_p}} \sum_{l=0}^{\frac{B_n^{max}}{\Delta_p}} \Pi_{bs,n}(j,l) = 1. \tag{27}
$$

However, the BS queue evolves independently of the UE queue, thus the marginal steady-state probability for the BS, $\Pi_{bs}(j)$, can be derived from the BS queue being modeled as a one-dimensional Markov chain where $Q_{bs}(i,j)$ is characterized as follows aligning to the conditions described earlier.

Condition (1): $i \leq \frac{b_{th}^{bs}}{\Delta_p}$

$$
\begin{aligned}
Q_{bs}((i,j)) &= Pr\left(j = i + \frac{E_{bs}^{H}}{\Delta_p}\right) \\
&= Pr\left(E_{bs}^{H} = (j - i)\Delta_p\right).
\end{aligned} \tag{28}
$$

Condition (2): $i > \frac{b_{th}^{bs}}{\Delta_p}$

$$
\begin{aligned}
Q_{bs}(i,j) &= Pr\left(j = i + \frac{E_{bs}^{H}}{\Delta_p} - \frac{E_{bs}^{C}}{\Delta_p}\right) \\
&= Pr\left(E_{bs}^{H} = (j - i + \frac{E_{bs}^{C}}{\Delta_p})\Delta_p\right).
\end{aligned} \tag{29}
$$

Similarly, in steady-state, the BS state-probability, $\Pi_{bs}(j)$, is expressed as follows using Markovian analysis,

$$
\Pi_{bs}(j) = \sum_{i=0}^{\frac{B_{bs}^{max}}{\Delta_p}} Q_{bs}(i,j) \times \Pi_{bs}(i), \tag{30}
$$

where

$$
\sum_{j=0}^{\frac{B_{bs}^{max}}{\Delta_p}} \Pi_{bs}(j) = 1. \tag{31}
$$

Since the UE energy harvesting depends on the BS state, the steady-state probability that the UE is in state l is the sum of the joint probabilities over all BS states,

$$
\Pi_n(l) = \sum_{j=0}^{\frac{B_{bs}^{max}}{\Delta_p}} \Pi_{bs,n}(j,l), \tag{32}
$$

IV. Performance Analysis

The aim of this work is to determine closed-form expressions for UE outage probability. The probability of UE outage is defined by the channel outage, which is the probability that the instantaneous SINR is less than some set system target SINR. With our based on the UE, let $k^{'}$ represent the target rate power, P_n^R, k represent the available battery power and UE maximum power be represented as, $B^p = \frac{B_n^{max}}{t_{ts}}$. Then, UE outage probability given $k^{'}$ is expressed as follows,

$$
\begin{aligned}
P_n^{out}(t)|_{k^{'}} &= Pr\left(k < k^{'}\right) \\
&= \sum_{k=0}^{k^{'}} \Pi_n(k),
\end{aligned} \tag{33}
$$

Unconditioning on $k^{'}$, the above Equation (33) reduces to the below expression representing the general expression for UE outage probability in the considered system model.

$$
P_n^{out}(t) = \sum_{k^{'}=0}^{\frac{B^p}{\Delta_p}} \left(\sum_{k=0}^{k^{'}} \Pi_n(k)\right) \Pi_n(k^{'}). \tag{34}
$$

V. Results and Discussions

In this section, we confirm the derived analytical results for the considered IoT device, UE_n, in our analysis. For results generation, we assume the IoT device is at the fixed distance of 100 meters with the pass loss exponent, (pl), set to 3.0. The uplink and downlink channel gain rates between the IoT device and the base-station are calculated as follows, $\lambda = d^{(-pl)}$, where d is the distance between the two nodes using a distance normalization factor of 100 meters. Additive White Gaussian Noise, $\sigma = 0.01$, $\Delta_p = 0.01$, $T_f = 20 \times 10^{-3}$seconds, $t_{ts} = 2 \times 10^{-3}$seconds, $B^{max} = 50$ joules (discritized), $\eta = 1$, simulation iterations to 1×10^6, bandwidth: $W = 5$MHz.

Fig.2 shows outage probability versus target SINR results for the considered UE for different values BS transmit power, $P^{bs} = \{40, 80, 120\}$ watts. The results show a close match between the simulation and analysis plots. Similarly, Fig.3 shows outage probability versus target SINR for different values of uplink UE target transmission rate, $R^t = \{128, 256, 384, 512\}$kbps. This also shows a close match between outage simulation results and analysis, thus confirming the correctness of the analytical approach adopted in this paper.

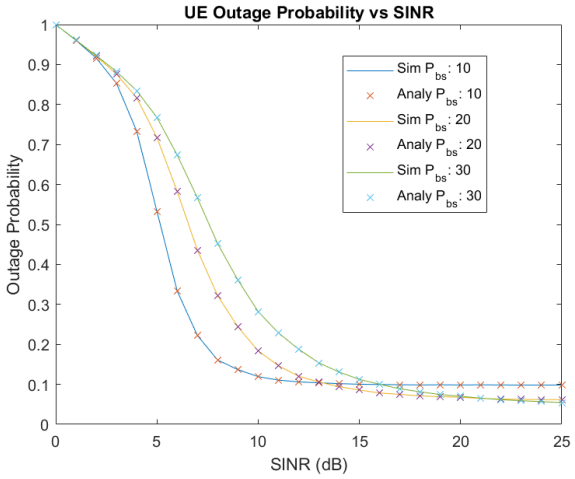


Fig. 2. UE_n Outage Probability vs SINR: $P_{bs} = \{10, 20, 30\}$ Watts

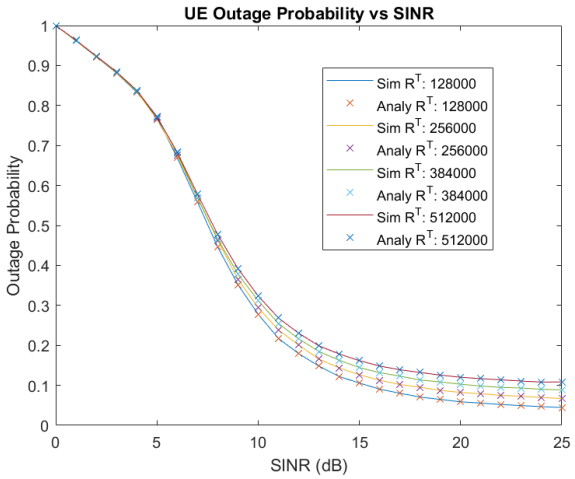


Fig. 3. UE_n Outage Probability vs SINR: $R^T = \{128, 256, 384, 512\}$ kbps

VI. Conclusion

In this paper, we model outage probability analysis for an IoT device operating in a green energy Network where the IoT device harvest RF energy from the BS and the BS harvests energy from an ambient source, in the considered case, it is solar energy. We use Markovian analysis approach to derive UE energy states at each transmission frame. The derived Markovian model is then used to derive closed-form expression for the considered IoT device outage probability as shown in Section IV of this paper. This analysis can be used as the basis for the design and deployment of green energy IoT energy networks.

References

[1] M. H. Alsharif, A. Jahid, A. H. Kelechi, and R. Kannadasan, "Green iot: A review and future research directions," *Symmetry*, vol. 15, no. 3, p. 757, 2023.

[2] R. Arshad, S. Zahoor, M. A. Shah, A. Wahid, and H. Yu, "Green iot: An investigation on energy saving practices for 2020 and beyond," *Ieee Access*, vol. 5, pp. 15667–15681, 2017.

[3] F. K. Shaikh, S. Zeadally, and E. Exposito, "Enabling technologies for green internet of things," *IEEE Systems Journal*, vol. 11, no. 2, pp. 983–994, 2017.

[4] A. Singh, S. Redhu, B. Beferull-Lozano, and R. M. Hegde, "Network-aware rf-energy harvesting for designing energy efficient iot networks," *Internet of Things*, vol. 22, p. 100770, 2023.

[5] P. Kamalinejad, C. Mahapatra, Z. Sheng, S. Mirabbasi, V. C. M. Leung, and Y. L. Guan, "Wireless energy harvesting for the internet of things," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 102–108, 2015.

[6] "Design architectures for energy harvesting in the internet of things," *Renewable and Sustainable Energy Reviews*, vol. 128, p. 109901, 2020.

[7] B. Zhang, C. Dong, M. El-Hajjar, and L. Hanzo, "Outage analysis and optimization in single- and multiuser wireless energy harvesting networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1464–1476, 2016.

[8] N. P. Le, "Outage probability analysis in power-beacon assisted energy harvesting cognitive relay wireless networks," *Wireless Communications and Mobile Computing*, vol. 2017, no. 1, p. 2019404, 2017.

[9] W. Li, M.-L. Ku, Y. Chen, and K. J. Ray Liu, "On outage probability for two-way relay networks with stochastic energy harvesting," *IEEE Transactions on Communications*, vol. 64, no. 5, pp. 1901–1915, 2016.

[10] A. H. Sakr and E. Hossain, "Analysis of k-tier uplink cellular networks with ambient rf energy harvesting," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2226–2238, 2015.

[11] X. Zhang, K. Xiong, W. Chen, P. Fan, B. Gao, and K. B. Letaief, "Maximizing harvested energy in natural energy powered rf wpt with nonlinear eh model," *IEEE Transactions on Wireless Communications*, 2025.

[12] N. Kumar, C. Shekhar, and R. K. Tiwari, "Congestion analysis of finite tandem queueing network," in *2023 3rd International Conference on Advancement in Electronics & Communication Engineering (AECE)*, pp. 997–1001, IEEE, 2023.

[13] L. Le and E. Hossain, "Tandem queue models with applications to qos routing in multihop wireless networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 8, pp. 1025–1040, 2008.

[14] A. Brandwajn and Y.-L. L. Jow, "An approximation method for tandem queues with blocking," *Operations research*, vol. 36, no. 1, pp. 73–83, 1988.

[15] H. G. Perros and T. Altiok, "Approximate analysis of open networks of queues with blocking: Tandem configurations," *IEEE Transactions on Software Engineering*, vol. SE-12, no. 3, pp. 450–461, 1986.

[16] R. Atat, R. S. Bradley, and E. Serpedin, "Toward solar energy harvesting for small cell networks: Technology, feasibility, and challenges," in *2019 5th Experiment International Conference (exp. at'19)*, pp. 90–95, IEEE, 2019.

[17] H. Wang, M. Huang, Z. Zhao, Z. Guo, Z. Wang, and M. Li, "Base station wake-up strategy in cellular networks with hybrid energy supplies for 6g networks in an iot environment," *IEEE internet of things journal*, vol. 8, no. 7, pp. 5230–5239, 2020.

[18] R. Atat, L. Liu, J. Wu, J. Ashdown, and Y. Yi, "Green massive traffic offloading for cyber-physical systems over heterogeneous cellular networks," *Mobile Networks and Applications*, vol. 24, no. 4, pp. 1364–1372, 2019.

[19] A. Mostowski, "Sc kleene. recursive functions and intuitionistic mathematics. proceedings of the international congress of mathematicians, cambridge, massachusetts, usa, august 30-september 6, 1950, american mathematical society, providence1952, vol. i, pp. 679–685.," *The Journal of Symbolic Logic*, vol. 18, no. 2, pp. 181–182, 1953.

[20] A. Wyner, "Recent results in the shannon theory," *IEEE Transactions on information Theory*, vol. 20, no. 1, pp. 2–10, 1974.
[21] G. Pan, H. Lei, Y. Yuan, and Z. Ding, "Performance analysis and optimization for swipt wireless sensor networks," *IEEE Transactions on Communications*, vol. 65, no. 5, pp. 2291–2302, 2017.
[22] K. Patil, K. De Turck, and D. Fiems, "A two-queue model for optimising the value of information in energy-harvesting sensor networks," *Performance Evaluation*, vol. 119, p. 27–42, Mar 2018.

Evaluation Framework for LMS-Based Channel Estimation in MIMO-NOMA Systems

Lehlohonolo Sekokotoana
School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg, South Africa
1918749@students.wits.ac.za

Fambirai Takawira
School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg, South Africa
fambirai.takawira@wits.ac.za

Olutayo Oyerinde
School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg, South Africa
olutayo.oyerinde@wits.ac.za

***Abstract*—This paper presents a unified performance evaluation framework for a least mean square (LMS)-based channel estimation (CE) approach for a single cell multiple-input-multiple-output non-orthogonal multiple access (MIMO-NOMA) system. This is advanced by recasting the simplified 2×2 uplink (UL) MIMO-NOMA model to a multiple-input-single-output (MISO)-NOMA model, and subsequently, the CE error model is established. The focus, therefore, is the characterisation of this system's CE performance using closed-form analytical expressions of the asymptotic CE error. Through Monte-Carlo simulations, the LMS-based CE scheme is assessed under the proposed performance evaluation framework, using the mean square error (MSE) as the metric. Overall, the LMS-based CE scheme shows adequate MSE performance in terms of CE accuracy and convergence rate. Furthermore, simulation results tally with the theoretical analysis.**

Keywords—Channel estimation, least mean squares (LMS), mean square error (MSE), NOMA, MIMO-NOMA.

I. INTRODUCTION

One of the key enabling technologies for the fifth generation (5G) of wireless mobile networks (and beyond) is multiple-input-multiple-output (MIMO), specifically, massive MIMO (MaMIMO). The underlying concept of MIMO involves the employment of multiple antennas at the transmitter and receiver ends to improve the overall spectral efficiency of the communication link [1–4]. Power-domain non-orthogonal multiple access (NOMA) is another candidate technology for 5G networks and beyond, which also improves the overall spectral efficiency by enabling multiple user access through superimposing multiple user signals with different power levels on the same time and frequency resources.

Combining the two principles, MIMO-NOMA systems have been shown to significantly improve capacity and coverage. Nevertheless, the practical realisation of the aforementioned performance gains critically relies on reliable communication, which requires accurate information recovery at the receiver. This necessitates precise channel estimation (CE), a vital signal processing principle, to guarantee the success of communication systems [5–7].

The importance of practical CE schemes in MIMO-NOMA systems has been acknowledged by the work presented in [5–9]. In their work, Cheng et al. [7] advance a novel pilot design strategy for channel state information (CSI) acquisition in downlink (DL) NOMA-based multiuser (MU) MIMO systems. For this purpose, the proposed training scheme is evaluated under different practical scenarios, with conventional estimators such as the minimum mean square error (MMSE) employed for demonstrating the effects of CE. Although the study in [7] is not centred around CE, the key findings highlight the susceptibility of NOMA to inter-group interference, and the requirement of accurate channel estimates at the users for seamless operation.

In contrast to [7], Hu et al. [9] advance a more accurate CE evaluation framework, adopting the normalised MSE (NMSE) as the metric. Particularly, they propose a semi-blind CE method for multi-cell MaMIMO-NOMA using an uncommon group SIC (GSIC) decoding technique to mitigate pilot contamination. The novel CE scheme recovers the uplink (UL) channel parameters on a group-by-group basis and shows superior NMSE performance compared to the conventional least squares (LS) and MMSE counterparts. Nonetheless, [9] is limited to uplink transmission assuming single-antenna users. Additionally, the asymptotic MSE analysis is not considered in the performance evaluation presented [9].

More recently, emerging works have explored CE application in numerous variants of NOMA-based systems [10–14]. A joint CE and power allocation coefficients approximation problem for multiple-input-multiple-output (MISO)-NOMA systems is addressed in [10]. As a solution, [10] advances a hybrid scheme that employs the long-short-term-memory (LSTM) network and the convolutional neural network (CNN) for CE and power allocation, respectively. The work in [10] reports superior performance over the MMSE estimator benchmark for CE, and improved bit error rate (BER) performance compared to the analytical counterpart for power allocation. The hybrid scheme was, however, only evaluated using BER and outage probability instead of the MSE.

The remaining studies on NOMA-based systems focus on CE applications in specific contexts, including millimetre-wave (mmWave) MaMIMO systems, cell-free and grant-free MaMIMO systems, which are beyond the scope of the work proposed herein [11–14]. Nevertheless, one of the recurring challenges identified in the covered work is the application for single-antenna users, and the missing asymptotic MSE analysis. Furthermore, the CE approaches covered are typically attributed

with high computational complexity, which could be alleviated by the employment of low-complexity algorithms such as the LMS.

To this end, this paper proposes a performance evaluation framework for a low complexity least mean square (LMS)-based CE scheme for power-domain (PD) MIMO-NOMA systems in UL transmission scenarios. The major contribution of this paper is in the derivation of the analytical expression for the CE mean square error (MSE). An account that presents a low complexity CE scheme for MIMO-NOMA applications is presented in [15], investigating the usage of the recursive LS (RLS) and variable forgetting factor RLS (VFFRLS) algorithms. Although their proposed VFFRLS was shown to have performance gains over the RLS, the authors of [15] do not characterise the analytical error variance in their performance evaluation.

In literature, the closed-form derivation of the CE error variance is well documented for the LMS algorithm and other closely related adaptive algorithms, such as the Kalman and Weiner filter variants, predominantly in orthogonal multiple access (OMA) scenarios. Although focused on single-input-single-output (SISO) scenarios, some of these accounts can be referred to in the work presented in [6] and [16]. Extensions to MIMO scenarios are suggested in the referred accounts, acknowledging the potential performance gains thereof. While the work in [16] presents CE for an OMA system, the study in [6] considers a DL SISO-NOMA transmission scenario.

Derivations of analytical CE MSE specific to MIMO systems employing adaptive algorithms are presented in papers [17–19]. A performance evaluation framework for an unrestricted block LMS (UBLMS) CE algorithm is established in [17]. This is advanced by the examination of the conventional MSE and the complementary MSE (CMSE) of the block LMS (BLMS) algorithm for both the transient and steady-state stages. Moreover, the derivations of closed-form expressions for the MSE and the CMSE were outlined, and simulation results were shown to verify the theoretical analysis. The findings in [17] provide a basis for the design of frequency-domain LMS-based CE approaches considering noncircular signals.

In [18], Singh et al. present LMS and BLMS CE approaches for MU-MIMO systems in their work. By performing CE at the BS, the CE approaches in [18] are documented to introduce a simplified receiver design. Consequently, the proposed schemes exhibit significant performance gains relative to their Kalman-filter counterparts, and they are also shown to verify the theoretical analysis without the requirement of a priori information. Other similarly themed literature can be referred to in [19–23], which cover a variety of systems that have similar modelling to MIMO systems.

The most obvious limitation of the work presented in [16–23] is their adaptation to none-NOMA (i.e., OMA) systems – the reported performance gains may not necessarily apply to NOMA scenarios. In this paper, the theoretical characterisation of the CE performance of a multi-antenna MIMO-NOMA system is advanced. The main contribution of this paper is the derivation of closed-form expressions of the CE error for the recovery of the UL channel between each user equipment (UE) and the BS. To this end, a two-level Taylor expansion approximation of the magnitude factors in the asymptotic MSE terms is effected, leveraging the fading assumptions used in the derivations presented in [5] and [6] for the tractability of the theoretical analysis. Furthermore, the expressions for the optimal step-size of the LMS algorithm, and consequently, the corresponding minimal error variance, are derived. The CE error variance is used as the benchmark performance for the LMS algorithm at steady state.

II. System Model

A. Channel Model

An uplink (UL) MIMO-NOMA system that consists of a base-station (BS) serving two users is considered. The BS is equipped with M_r antennas while each of the two users, $UE_i, \forall i = 1,2$, are equipped with M_t antennas whereby $M_r \geq M_t$. Furthermore, the two users are assumed to have significantly different channel conditions, with the first user, UE_1, located farther away from the BS (thus UE_1 experiences a higher path loss) relative to UE_2. With this setting, the DL channel between the BS and each user UE_i can be modelled as follows [1 – 4]:

$$\boldsymbol{G}_{i,n}^{H} = \sqrt{L_i}\boldsymbol{H}_{i,n}^{H}, \tag{1}$$

where $\boldsymbol{H}_{i,n}^{H} = \begin{bmatrix} h_{i,1,1,n} & \cdots & h_{i,1,M_r,n} \\ \vdots & \ddots & \vdots \\ h_{i,M_t,1,n} & \cdots & h_{i,M_t,M_r,n} \end{bmatrix} \in \mathbb{C}^{M_t \times M_r}$, and $h_{i,m_t,m_r,n} \sim CN\left(0, \sigma^2_{h_{i,m_t,m_r}}\right)$, is the complex Rayleigh fading channel gain with $m_r(m_t) \in \{1,..,M_r(M_t)\}$. The deterministic distance-related attenuation factor, $L_i > 0$, such that $L_1 > L_2$, is assumed to be known to both the users and the BS. Jakes' Doppler spectrum is also assumed in modelling the slow-fading channel coefficients, $h_{i,m_t,m_r,n}$, and the attributed autocorrelation is expressed as:

$$\begin{aligned} R_{h_{i,m_t,m_r,n}}(m) &= E\left\{h_{i,m_t,m_r,n} \cdot \left(h_{i,m_t,m_r,n-m}\right)^*\right\} \\ &= \sigma^2_{h_{i,m_t,m_r}} J_0(2\pi f_d T \cdot m), \end{aligned} \tag{2}$$

where m is the time lag, and the corresponding Doppler spectrum of the element $h_{i,m_t,m_r,n}$ in (2) is:

$$\Gamma_{h_{i,m_t,m_r,n}}(f) = \begin{cases} \dfrac{\sigma^2_{h_{i,m_t,m_r}}}{\pi f_d \sqrt{1 - \left(\dfrac{f}{f_d}\right)^2}} & if\ |f| < f_d \\ 0 & if\ |f| > f_d \end{cases}, \tag{3}$$

where $f = f_d \cos\varphi$ is the frequency of the arriving signal, f_d is the maximum Doppler frequency, φ is the phase of arrival uniformly distributed within $[0\ 2\pi)$, and $\sigma^2_{h_{i,m_r,m_t}}$ is the variance of the channel coefficients. On the other hand, the uplink channel between UE_i and the BS is modelled as [1 – 3]:

$$\boldsymbol{G}_{i,n} = \sqrt{L_i}\boldsymbol{H}_{i,n} \in \mathbb{C}^{M_r \times M_t}, \tag{4}$$

assuming the same model considerations as described for the DL counterpart.

B. Signal Model

Since the proposed CE scheme is training-aided, we focus on the pilot transmission in the UL scenario. To this end, each user UE_i transmits training data described as follows:

$$\boldsymbol{x}_{i,n} = \sqrt{\alpha_i P}\boldsymbol{s}_{i,n}, \tag{5}$$

$$\boldsymbol{s}_{i,n} = \left[s_{i,1,n}, \dots, s_{i,M_t,n}\right]^T, \tag{6}$$

where $\boldsymbol{x}_{i,n}$ is a transmitted pilot signal vector, $\boldsymbol{s}_{i,n}$ is the corresponding pilot symbol vector, α_i is the power allocation coefficient for user UE_i following $\sum_{i=1}^{2}\alpha_i \leq 1$, P is the total transmit (Tx) power, and n denotes the time index. Furthermore, each symbol, $s_{i,m_t,n}$, is assumed to be normalized, i.e., $E\left\{\left|s_{i,m_t,n}\right|^2\right\} = 1$ and each pilot signal element, $x_{i,m_t,n}$, follows $E\left\{\left|x_{i,m_t,n}\right|^2\right\} \approx M_t\alpha_i P$. With this setting, the BS receives pilot signals from all the users, modelled as the following observation:

$$\boldsymbol{y}_n = \sum_{i=1}^{2}\boldsymbol{G}_{i,n}\boldsymbol{x}_{i,n} + \boldsymbol{w}_n, \tag{7}$$

where $\boldsymbol{y}_n = \left[y_{1,n}, \dots, y_{M_r,n}\right]^T$ is the observation vector, $\boldsymbol{w}_n = \left[w_{1,n}, \dots, w_{M_r,n}\right]^T$ is the additive white Gaussian noise (AWGN) vector with elements following the distribution $w_{m_r,n} \sim CN(0, \sigma_w^2)$. Since the BS observes the pilot signals from all the users as a composite signal $\boldsymbol{y}_n$, the pilot signal vectors, $\boldsymbol{x}_{i,n}$, for each user UE_i should be selected such that they are orthogonal (i.e., $\left(\boldsymbol{x}_{1,n}\right)^T\boldsymbol{x}_{2,n} = 0$) [5,6]. This configuration ensures the feasibility of uniquely identifying the pilot signal from each user.

III. Proposed LMS-Based CE Scheme

A. Problem Formulation & Evaluation Framework

Given the UL-MIMO-NOMA as defined in section I, CE at the BS entails the recovery of the channel matrix $\boldsymbol{H}_{i,n}$, which is comprised of the small-scale fading channel parameters, $h_{i,m_r,m_t,n}$. In addressing this estimation problem, the observation model in (7) is decomposed into its antenna-wise elements, $y_{m_r,n}, \forall m_r \in \{1, \dots, M_r\}$, which are assessed separately. This effectively reduces the system into two independent MISO-NOMA configurations – valid assuming the antenna channel links are spatially uncorrelated. The observation model associated with each Rx antenna m_r is therefore expressed as follows:

$$y_{m_r,n} = \sum_{i=1}^{2}\sqrt{L_i}\boldsymbol{h}_{i,m_r,n}\boldsymbol{x}_{i,n} + w_{m_r,n}, \tag{8}$$

where $\boldsymbol{h}_{i,m_r,n} = \left[\boldsymbol{H}_{i,n}\right]_{m_r,:} \in \mathbb{C}^{1\times M_t}$, and $\lfloor\boldsymbol{A}\rfloor_{n,:}$ denotes n-th row of the matrix $\boldsymbol{A}$. Expanded into its explicit summation form, the observation in (8) becomes (for a 2×2 MIMO system):

$$y_{m_r,n} = \sum_{i=1}^{2}\sqrt{L_i}\left(h_{i,m_r,m_t,n}x_{i,m_t,n} + h_{i,m_r,\sim m_t,n}x_{i,\sim m_t,n}\right) + w_{m_r,n}, \tag{9}$$

where the *tilde* notation, ~, designates the complementary Tx antenna index, i.e., $\sim m_t$ denotes the Tx antenna index *other than* m_t.

This new MISO-NOMA representation serves as the input to the adaptive LMS algorithm, outlined in the ensuing section. To assess the CE performance of the LMS-based scheme, the proposed evaluation framework involves the development of an analytical model for the CE error variance. In effect, this entails the derivation of the closed-form asymptotic CE error expressions. As such, the CE error is defined as follows:

$$\boldsymbol{e}_{i,m_r,n} = \boldsymbol{h}_{i,m_r,n} - \hat{\boldsymbol{h}}_{i,m_r,n}, \tag{10}$$

where $\hat{\boldsymbol{h}}_{i,m_r,n}$ is the estimated channel coefficient vector at Rx antenna, m_r, and $\boldsymbol{e}_{i,m_r,n}$ is the corresponding CE error vector, with $\boldsymbol{e}_{i,m_r,n} = \left[e_{i,m_r,1,n}, \dots, e_{i,m_r,M_t,n}\right]$ and $e_{i,m_r,m_t,n} = h_{i,m_r,m_t,n} - \hat{h}_{i,m_r,m_t,n}$. The error variance vector associated with (10) is defined as shown in the following expression:

$$\boldsymbol{\sigma}_{i,e}^2 \triangleq \begin{pmatrix} E\left\{\left|e_{i,m_r,1,n}\right|^2\right\} \\ \vdots \\ E\left\{e_{i,m_r,M_t,n}{}^2\right\} \end{pmatrix}. \tag{11}$$

For the individual Tx and Rx antenna pair, i.e., each element (11), of the error variance is expressed as:

$$\sigma_{i,e}^2 \triangleq E\left\{\left|e_{i,m_r,m_t,n}\right|^2\right\}. \tag{12}$$

B. LMS Algorithm

In this section, the LMS algorithm employed previously in [9,10] is adapted for each MISO-NOMA UL system model. At the BS, the chosen LMS cost function is given by the following expression:

$$J_{i,m_r}\left(\hat{\boldsymbol{h}}_{i,m_r,n}\right) = E\left\{\left|y_{m_r,n} - \sum_{i=1}^{2}\sqrt{L_i}\,\hat{\boldsymbol{h}}_{i,m_r,n}\boldsymbol{x}_{i,n}\right|^2\right\},$$

$$\approx \left|y_{m_r,n} - \sum_{i=1}^{2}\sqrt{L_i}\,\hat{\boldsymbol{h}}_{i,m_r,n}\boldsymbol{x}_{i,n}\right|^2, \tag{13}$$

where $\hat{\boldsymbol{h}}_{i,m_r,n}\boldsymbol{x}_{i,n} = \sum_{i=1}^{2}\sqrt{\alpha_i P}\sum_{m_t=1}^{M_t}\hat{h}_{i,m_r,m_t,n}\,s_{i,m_t,n}$. For analysis, a simplified pilot transmission case of $s_{i,m_t,n} = 1$, $\forall m_t \in \{1, \dots, M_t\}$, is considered. With this basis, we consider the standard LMS estimator update rule, which is defined as:

$$\hat{\boldsymbol{h}}_{i,m_r,n+1} = \hat{\boldsymbol{h}}_{i,m_r,n} - \frac{1}{2}\mu\frac{d\left[J_{i,m_r}\left(\hat{\boldsymbol{h}}_{i,m_r,n}\right)\right]}{d\hat{\boldsymbol{h}}_{i,m_r,n}}, \tag{14}$$

where μ is the learning rate subject to $0 < \mu < \frac{M_t\sum_{i=1}^{2}L_i\alpha_i P}{SNR}$. Without loss of generality, a 2×2 MIMO-NOMA system is considered for analysis. Maintaining the data model as recast to a MISO-NOMA scenario, shown in (9), a gradient descent is performed on the cost function to find the update rule. The channel coefficient estimates at each Rx antenna m_r from the Tx antenna m_t, $\forall m_t$ and $\forall m_r \in \{1,2\}$, are therefore generalised by the following update rule:

$$\hat{h}_{i,m_r,m_t,n+1} = \hat{h}_{i,m_r,m_t,n}$$

$$+\mu p_i\left(y_{m_r,n}-\sum_{j=1}^{2}p_j\sum_{m_t=1}^{2}\hat{h}_{j,m_r,m_t,n}\right), \quad (15)$$

where $p_i=\sqrt{L_i\alpha_i P}$, and $j\in\{1,2\}$.

Given the adaptability and iterative nature of the LMS algorithm, the extension of the proposed CE scheme to a scenario of more than two antennas is easily demonstrable: a generalized version of (15) for a scenario of Tx and Rx antennas such that $M_r \gg M_t > 2$, where M_t and M_r represent the total number of Tx and Rx antennas respectively, is expressed as $\hat{h}_{i,m_r,m_t,n+1}=\hat{h}_{i,m_r,m_t,n}+\mu p_i\big(y_{m_r,n}-\sum_{j=1}^{2}p_j\sum_{m_t=1}^{M_t}\hat{h}_{j,m_r,m_t,n}\big)$. The CE approach at the BS for each Rx antenna m_r is summarised by equation (15), for each of the complex channel gain elements in the channel estimate matrix $\widehat{\boldsymbol{H}}_{i,n}$.

C. Error Variance Analysis

This section covers the derivation of the closed-form expressions of the asymptotic error. Firstly, the update rule (15) for each Rx-Tx antenna pair considered in the z-domain is as shown below:

$$\widehat{H}_{i,m_r,m_t}(z)z=\widehat{H}_{i,m_r,m_t}(z)+\mu p_i\left(Y_{i,m_r}(z)-\sum_{j=1}^{2}p_j\sum_{m_t=1}^{M_t}\widehat{H}_{i,m_r,m_t}(z)\right). \quad (16)$$

To express the z-domain CE error expression, the z-domain expressions of the observation in (9) and the error given by (10), are substituted into (16), ensuring that the update rule is expressed in terms of the true channel, the channel estimation errors and the AWGN. After initial substitution, the following expression is obtained:

$$H_{i,m_r,m_t}(z)z-E_{i,m_r,m_t}(z)z=H_{i,m_r,m_t}(z)-E_{i,m_r,m_t}(z)+\mu p_i\left(\sum_{j=1}^{2}p_j\sum_{m_t=1}^{M_t}E_{j,m_r,m_t}(z)\right)+\mu p_i W_{m_r}(z). \quad (17)$$

The analysis is then advanced by considering the scenario whereby the channels corresponding to each transmit antenna are uncorrelated. With this configuration, $H_{i,m_r,m_t}(z)=H_{i,m_r,\sim m_t}(z)$ and $E_{i,m_r,m_t}(z)=E_{i,m_r,\sim m_t}(z)$. Applying this assumption, and after some algebraic manipulations, (17) becomes:

$$\begin{aligned}E_{i,m_r,m_t}(z)\left(1-(z+\mu p_i^2)\right)&=H_{i,m_r,m_t}(z)(1-z)+\mu p_i^2E_{i,m_r,m_t}(z)+\mu p_i W_{i,m_r}(z),\\ \Rightarrow E_{i,m_r,m_t}(z)\left(1-\left(z+2\mu p_i(p_i+p_{\sim i})\right)\right)&=H_{i,m_r,m_t}(z)(1-z)+\mu p_i W_{i,m_r}(z).\end{aligned} \quad (18)$$

From (18), the final CE error expression $E_{i,m_r,m_t}(z)$ is therefore simplified to:

$$E_{i,m_r,m_t}(z)=H_{i,m_r,m_t}(z)A_i(z)+W_{i,m_r}(z)B_i(z), \quad (19)$$

where $A_i(z)=\frac{(z-1)}{\left(z-(1-2\mu p_i(p_i+p_{\sim i}))\right)}$ and $B_i(z)=\frac{-\mu p_i}{\left(z-(1-2\mu p_i(p_i+p_{\sim i}))\right)}$. Using the error variance definition given in (12) and the error expression in (19), the frequency domain steady-state MSE for user i, $\forall i\in\{1,2\}$ is expressed as:

$$\sigma_{e,i}^2=MSE1_i+MSE2_i, \quad (20)$$

where $MSE1_i\approx\int_{-f_d}^{f_d}|A_i(f)|^2\Gamma_{h_{i,m_r,m_t}}(f)\,df$, $MSE2_i=\int_{\frac{-1}{2T}}^{\frac{1}{2T}}|B_i(f)|^2\Gamma_w(f)\,df$, and the bandlimited noise PSD is denoted $\Gamma_{w_{m_r}}(f)=\begin{cases}\sigma_w^2T, & |f|<\frac{1}{2T}\\ 0, & Otherwise\end{cases}$.

D. Asymptotic Error Closed-form Expression

To derive the closed-form expressions for the asymptotic MSE, two levels of approximations are effected for the first integrand term factor of (20). By virtue of the slow-fading assumptions [5,6], the first-level approximation is realised by expressing z as the first two terms of its Taylor expansion, i.e., $z=e^{2j\pi fT}\approx 1+j2j\pi fT$. Consequently, the absolute square expression is evaluated using the product of the expression and its conjugate, i.e., $|A_i(f)|^2\approx\big(A_i(f)\big)\big(A_i(f)\big)^*$; where $(\cdot)^*$ is the conjugate operation. To this end, the square magnitude $|A(f)|^2$ is evaluated as follows:

$$\left|A_i(e^{2j\pi fT})\right|^2=\frac{(2j\pi fT)^2}{(2a)^2+(2j\pi fT)^2}. \quad (21)$$

Evaluating the first term integral in (20) is not tractable given the product of the expression in (21) and Jakes' power spectral density (PSD) expression in (3). The second-level approximation addresses this complexity by performing the third-order Taylor expansion of the magnitude expression given in (21) to arrive at the final expression given by (22):

$$\left|A_i(e^{2j\pi fT})\right|^2\approx\frac{(\pi fT)^2}{\left(\mu p_i(p_i+p_{\sim i})\right)^2}. \quad (22)$$

Given (22), term $MSE1_i$ in (20) is approximated as, $MSE1_i\approx\int_{\frac{-1}{2T}}^{\frac{1}{2T}}\left(\frac{(\pi fT)^2}{\mu p_i(p_i+p_{\sim i})}\right)\frac{\sigma_{h_{i,m_r,m_t}}^2}{\pi\sqrt{{f_d}^2-f^2}}\,df$, for $\forall m_t\in\{1,2\}$. The evaluation of $MSE1_i$ is simplified by introducing a variable change, as informed by the PSD expression in (3), using the relationships $f=f_d\cos\varphi$ and $f_d=\frac{1}{2T}$. This introduces the following upper and lower limits, respectively: $\varphi=cos^{-1}\left(\frac{0}{f_d}\right)=\frac{\pi}{2}$, and, $\varphi=cos^{-1}\left(\frac{f_d}{f_d}\right)=0$. As such, $MSE1_i$ is evaluated to the following final expression:

$$MSE1_i\approx\sigma_{h_i}^2\left(\frac{(\pi Tf_d)^2}{2\left(\mu p_i(p_i+p_{\sim i})\right)^2}\right). \quad (23)$$

The second term is first simplified to the form: $MSE2_{m_t}=\sigma_{w_{m_r}}^2T\int_{\frac{-1}{2T}}^{\frac{1}{2T}}|B(f)|^2df$. To evaluate the integral in $MSE2_{m_t}$, the

integrand expression first needs to be determined. Taking $C_i = \left(1 - 2\mu p_i(p_i + p_{\sim i})\right)$, the simplified expression of the integrand is obtained as:

$$|B_i(f)|^2 = \frac{(\mu p_i)^2}{(1-2C\cos(2\pi fT)+(C_i)^2)}. \quad (24)$$

Following some algebraic manipulations, the absolute square expression in (24) can further be simplified by introducing the following substitutions, $D_i = \frac{(\mu p_i)^2}{(1+(C_i)^2)}$ and $E_i = \frac{-2C_i}{(1+(C_i)^2)}$. With this setting, the simplified integrand expression is as follows:

$$|B_i(f)|^2 = \frac{D_i}{(1+E_i\cos(2\pi fT))}. \quad (25)$$

The simplified $MSE2_i$ expression is therefore given as:

$$MSE2_i = \sigma_w^2 T D_i \int_{\frac{-1}{2T}}^{\frac{1}{2T}} \left(\frac{1}{1+E_i\cos 2\pi fT}\right) df$$
$$= 2\sigma_w^2 T D_i \int_0^{\frac{1}{2T}} \left(\frac{1}{1+E_i\cos 2\pi fT}\right) df. \quad (26)$$

Let $v = 2\pi fT$, to simplify (26) by introducing a variable substitution, then $dv = 2\pi T df$ and the limits $\left[0\ \frac{1}{2T}\right]$ become $[0\ \pi]$. The integral in (26) therefore becomes:

$$MSE2_i = \frac{\sigma_w^2 D_i}{\pi}\int_0^{\pi} \frac{dv}{1+E_i\cos v}. \quad (27)$$

From section 3.613 in the table of integrals [23], the evaluation of the noise term in (27) gives the following expression:

$$MSE2_i = \frac{\sigma_w^2 D_i}{\sqrt{1-(E_i)^2}}, \quad (28)$$

where $(E_i)^2 < 1$, which is easily demonstrable given the expression of E_i. Substituting the expressions for C_i and D_i back into (28), the final expression for the second MSE term is expressed as:

$$MSE2_i = \sigma_w^2 \frac{\mu^2 p_i^2}{1-\left(1-2\mu p_i(p_i+p_{\sim i})\right)^2}$$
$$= \sigma_w^2 \frac{\mu^2 p_i^2}{1-\left(1-4\mu p_i(p_i+p_{\sim i})+4\left(\mu p_i(p_i+p_{\sim i})\right)^2\right)}$$
$$= \sigma_w^2 \frac{\mu^2 p_i^2}{4\mu p_i(p_i+p_{\sim i})\left(1-\mu p_i(p_i+p_{\sim i})\right)},$$
$$= \sigma_w^2 \frac{\mu p_i}{4(p_i+p_{\sim i})\left(1-\mu p_i(p_i+p_{\sim i})\right)}. \quad (29)$$

E. Optimal Step-Size & Minimum Error Variance

Given the final expressions in (23) and (29), the asymptotic MSE equation in (20) has been fully defined. The analytical expression for the optimal step-size, $\mu_{opt,i}$, can therefore be obtained by minimising $\sigma_{e,i}^2$ as follows:

$$\min_{\mu} \sigma_{e,i}^2. \quad (30)$$

For tractability, the following constraint is assumed to hold for sufficiently small step-sizes: $2\pi f_d T \ll \mu p_i(p_i+p_{\sim i}) \ll 1$, allowing for the following approximation, $\left(1-\mu p_i(p_i+p_{\sim i})\right) \approx 1$, to simplify the expression of $MSE2_i$. With this setting, the new error variance expression becomes:

$$\hat{\sigma}_{e,i}^2 = \sigma_{h_i}^2\left(\frac{(\pi f_d T)^2}{2\left(\mu p_i(p_i+p_{\sim i})\right)^2}\right) + \sigma_w^2\frac{\mu p_i}{4(p_i+p_{\sim i})}. \quad (31)$$

Therefore, the optimal step-size expression concerned with the estimation of the UL channel between the BS and $UE_i, \forall i\{1,2\}$, is expressed as:

$$\mu_{opt,i} = p_i\left(\frac{\sigma_{h_i}}{\sigma_w}\right)^{\frac{2}{3}}\frac{(2\pi f_d T)^{\frac{2}{3}}}{(p_i+p_{\sim i})^{\frac{1}{3}}}, \quad (32)$$

and the corresponding minimum error variance can be found by substituting $\mu_{opt,i}$ into the expression in (32),

$$\hat{\sigma}_e^2(\mu_{opt,i}) = \frac{3}{8}(2\pi f_d T)^{\frac{2}{3}}(\sigma_{h_i})^{\frac{2}{3}}\left(\frac{\sigma_w}{(p_i+p_{\sim i})}\right)^{\frac{4}{3}}. \quad (33)$$

IV. Results & Discussion

TABLE I. Simulation Parameters for the LMS-Based Approach

Parameter	*User 1 Value*	*User 2 Value*
Power Allocation Coefficient	0.25	0.75
Normalised Doppler Frequency	5×10^{-4}	5×10^{-4}
Distance Related Factor L_i	3	1.5
Step-size (μ) range	5×10^{-3} *to* 1.6 $\times10^{-1}$	5×10^{-3}*to* 3.58 $\times10^{-1}$
SNR	$20dB$	$20dB$

This section presents the results of the simulations conducted in applying the proposed performance evaluation framework adopting the LMS-based CE scheme. To this effect, the simulation setup considers the pilot transmission phase, wherein each user transmits pilots to the BS. Practically speaking, the MSE CE performance (for retrieving the channel estimates of a particular user) is virtually the same at both Rx antennas at the BS, as will be shown in the convergence characteristics. For the rest of the results, comparing the analytical and simulation MSE's , are only presented for one Rx antenna. Table I. summarises the parameters used in the simulation.

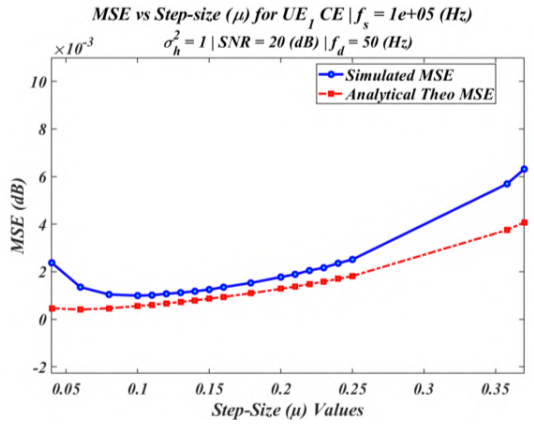


Fig. 1. Comparison of the analytical and simulated MSE performance over varying step-sizes (estimation of user 1's channel).

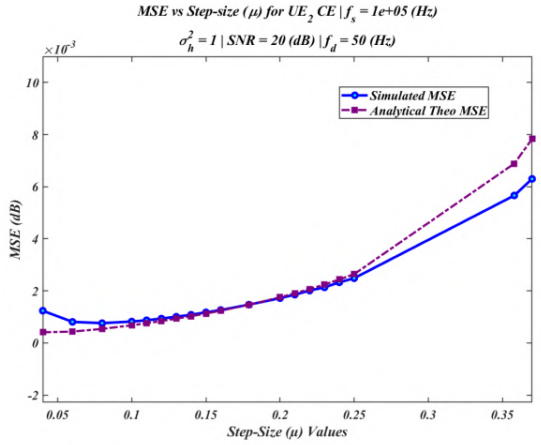


Fig. 2. Comparison of the analytical and simulated MSE performance over varying step-sizes (estimation of user 2's channel).

Fig. 1 and Fig. 2 investigate the CE MSE performance over a range of step-sizes for estimating the channels from UE_1 and UE_2 respectively. For both UEs, the simulation performance confirms the theoretical analysis carried out in Section III for the entire range of step-sizes considered. However, for step-sizes $\mu \leq 0.05$ and $\mu > 0.25$, the simulation MSE diverges. Analytical expressions for the upper and lower step-size limits have been derived, but have not been included in this paper for brevity.

In addition, the CE performance for UE_2 shows better MSE performance in comparison to its NOMA counterpart, owing to the configured power allocations and respective distance-related coefficients. For subsequent results, the step-size $\mu = 0.25$ is employed for the convergence characteristics of the proposed LMS algorithm, as presented in Fig. 3 and Fig. 4.

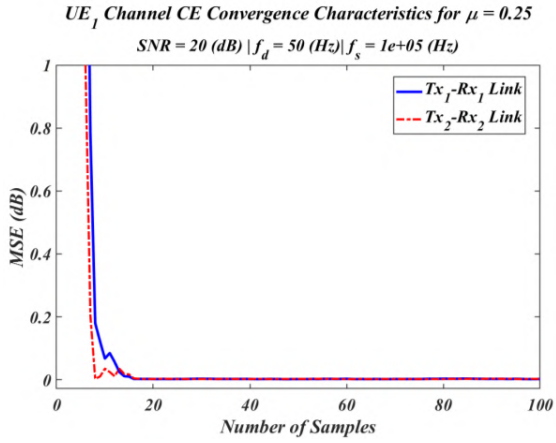


Fig. 3. Convergence characteristics of the LMS-based scheme (estimation of user 1's channel).

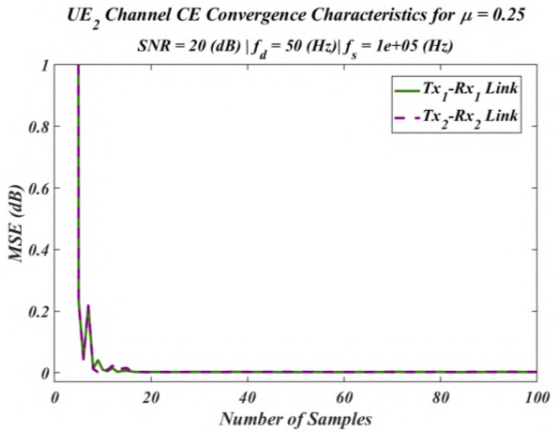


Fig. 4. Convergence characteristics of the LMS-based scheme (estimation of user 2's channel).

In Fig. 4, the convergence trends for the Rx 1 and Rx 2 at the BS are identical for each adopted learning rate, validating the assumed performance evaluation framework. Furthermore, the MSE performance accuracy achieved in Fig. 3 maintains the corresponding MSE accuracy in the simulation curves in Fig. 1 and Fig. 5, which is discussed in the following paragraph. With the step-size employed, CE performance for recovering user 2's channel exhibits superior MSE performance in terms of convergence speed and CE accuracy owing to the respective power allocations and large-scale coefficients. These trends assert the observations made for the results shown in both Fig. 2 and Fig. 4, respectively.

The MSE performance of the LMS-based CE approach is further evaluated for varying SNR conditions in Figures 5 and 6 for each respective user channel. For the entire SNR range considered, the simulation MSE corroborates the theoretical

analysis at both users employing optimal step-sizes. Furthermore, the simulation MSE performance in Fig. 4 and Fig. 5 maintains the CE MSE performance accuracy in Fig. 1, Fig. 2, Fig. 3, and Fig. 4 for the estimation of respective user channels at the BS.

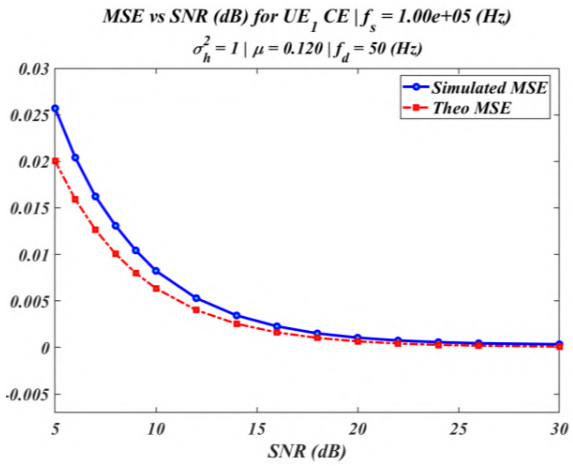


Fig. 5. Comparison of the analytical and simulated MSE performance in varying SNR conditions (estimation of user 1's channel).

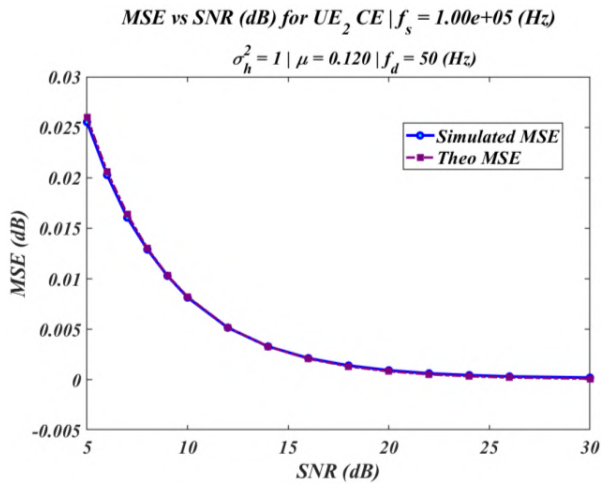


Fig. 6. Comparison of the analytical and simulated MSE performance performance in varying SNR conditions (estimation of user 2's channel).

V. Conclusion

In this paper, a performance evaluation framework for a low-complexity, pilot aided LMS-based CE approach for single cell MIMO-NOMA applications was introduced, providing a basis for system design. Adopting two-user UL 2×2 MIMO-NOMA system model, closed-form expressions of the error variance were derived using a two-level approximation technique to evaluate the asymptotic error expression. Furthermore, the optimal step-size and corresponding error variance expressions were derived. For demonstration purposes, the application of the LMS estimator for MIMO-NOMA scenarios with more than two antennas was also briefly presented. Overall, the analytical MSE was confirmed by the simulation MSE within the established step-size limits, and for all SNR values considered. Moreover, the MSE performance for the estimation of each user's channel was shown to largely depend on the large-scale coefficient under the proposed analysis. In practical terms, these insights provide a theoretical foundation for system design, whereby LMS step-sizes and NOMA power allocations can be fine-tuned according to large-scale coefficients for optimal CE performance. Future research can explore the proposed work for a DL scenario and a scenario with spatially correlated channel links. Alternatively, this performance evaluation framework can be implemented for mmWave-MaMIMO-NOMA scenarios.

References

[1] Flizikowski A, Marciniak T, Wysocki TA, Oyerinde O. Selected aspects of non orthogonal multiple access for future wireless communications. Math Comput Sci. 2023;17:1–29. https://doi.org/10.1007/s11786-023-00561-y. *(references)*

[2] Ding Z, Adachi F, Poor HV. Performance of MIMO-NOMA downlink transmissions. 2015 IEEE Glob Commun Conf GLOBECOM. San Diego, CA: IEEE; 2015. p. 1–6. https://doi.org/10.1109/GLOCOM.2015.7417060.

[3] Sun Q, Han S, I C-L, Pan Z. On the ergodic capacity of MIMO NOMA systems. IEEE Wirel Commun Lett. 2015;4:405–8. https://doi.org/10.1109/LWC.2015.2426709.

[4] Xiao C, Zeng J, Ni W, Su X, Liu RP, Lv T, et al. Downlink MIMO-NOMA for ultra-reliable low-latency communications. IEEE J Sel Areas Commun. 2019;37:780–94. https://doi.org/10.1109/JSAC.2019.2898785.

[5] Sekokotoana LE, Takawira F, Oyerinde OO. Least mean squares channel estimation for downlink non-orthogonal multiple access. 2019 IEEE Africon. Accra, Ghana: IEEE; 2019. p. 1–5. https://doi.org/10.1109/AFRICON46755.2019.9133868.

[6] Sekokotoana L, Takawira F, Oyerinde O. Successive interference cancelation-inspired channel estimation for downlink non-orthogonal multiple access. Trans Emerg Telecommun Technol. 2021;32:e4314. https://doi.org/10.1002/ett.4314.

[7] Cheng HV, Björnson E, Larsson EG. Performance analysis of NOMA in training-based multiuser MIMO systems. IEEE Trans Wirel Commun. 2018;17:372–85. https://doi.org/10.1109/TWC.2017.2767030.

[8] Kudathanthirige D, Amarasuriya G. Downlink training-based massive MIMO NOMA. ICC 2020 - 2020 IEEE Int Conf Commun ICC. Dublin, Ireland: IEEE; 2020. p. 1–6. https://doi.org/10.1109/ICC40277.2020.9149271

[9] Hu C, Wang H, Song R. Group successive interference cancellation assisted semi-blind channel estimation in multi-cell massive MIMO-NOMA systems. IEEE Commun Lett. 2021;25:3085–9. https://doi.org/10.1109/LCOMM.2021.3095119.

[10] Gaballa M, Abbod M, Alnasur S. Hybrid deep learning for channel estimation and power assignment for MISO-NOMA system. 2023 Int Conf IT Innov Knowl Discov ITIKD. 2023. p. 1–6. https://doi.org/10.1109/ITIKD56332.2023.10099781.

[11] Anu TS, Raveendran T. CNN-based Channel Estimation using NOMA for mmWave Massive MIMO System. 2023 IEEE Stat Signal Process Workshop SSP. 2023. p. 349–53. https://doi.org/10.1109/SSP53291.2023.10207968.

[12] Li S, Fang X, Dong H, Wu W, Liu F, Li Z. Cell-free massive MIMO-NOMA systems with nonreciprocal channels: performance analysis and calibration method design. IEEE Syst J. 2024;18:1729–40. https://doi.org/10.1109/JSYST.2024.3437769.

[13] Gollagi SG, Maheswari SS, Sapkale PV, Poojitha S. Channel estimation for pilot contamination in massive MIMO-NOMA system. J High Speed Netw. 2024;30:355–73. https://doi.org/10.3233/JHS-230043.

[14] Salari A, Shirvanimoghaddam M, Shahab MB, Arablouei R, Johnson S. Design and analysis of clustering-based joint channel estimation and signal detection for NOMA. IEEE Trans Veh Technol. 2024;73:2093–108. https://doi.org/10.1109/TVT.2023.3313650.

[15] Nayak A, Sarkar M, Sahoo S, Nanda S. Channel estimation of MIMO-NOMA system based on a variable forgetting factor recursive least square algorithm. 2021 IEEE 2nd Int Conf Appl Electromagn Signal Process Commun AESPC. 2021. p. 1–5. https://doi.org/10.1109/AESPC52704.2021.9708457.

[16] El Husseini AH, Simon EP, Ros L. Second-order autoregressive model-based kalman filter for the estimation of a slow fading channel described by the clarke model: optimal tuning and interpretation. Digit Signal Process. 2019;90:125–41. https://doi.org/10.1016/j.dsp.2019.04.008.

[17] Zhang X, Xia Y, Li C, Yang L, Mandic DP. Analysis of the unconstrained frequency-domain block LMS for second-order noncircular inputs. IEEE Trans Signal Process. 2019;67:3970–84. https://doi.org/10.1109/TSP.2019.2922149.

[18] Singh V, Srivastava S, Jagannatham AK. Superimposed pilots based adaptive time-selective channel estimation in MU-MIMO systems. 2020 Natl Conf Commun NCC. Kharagpur, India: IEEE; 2020. p. 1–6. https://doi.org/10.1109/NCC48643.2020.9056088.

[19] Srivastava S, Kumar MS, Mishra A, Chopra S, Jagannatham AK, Hanzo L. Sparse doubly-selective channel estimation techniques for OSTBC MIMO-OFDM systems: a hierarchical bayesian kalman filter based approach. IEEE Trans Commun. 2020;68:4844–58. https://doi.org/10.1109/TCOMM.2020.2995585.

[20] Zerguine A, Ahmad J, Moinuddin M, Al-Saggaf UM, Zoubir AM. An efficient normalized LMS algorithm. Nonlinear Dyn. 2022;110:3561–79. https://doi.org/10.1007/s11071-022-07773-0.

[21] Qadri SSU, Arif M, Saeed MOB. A novel variable step-size LMS algorithm for decentralized incremental distributed networks. Circuits Syst Signal Process. 2023;42:7226–49. https://doi.org/10.1007/s00034-023-02426-y.

[22] Fuyi Huang, ST Fan, Sheng Zhang, Hing Cheung So, Yang Jiang. Robust bias-compensated LMS algorithm: design, performance analysis and applications. IEEE Trans Veh Technol. 2023;72:13214–28. https://doi.org/10.1109/tvt.2023.3276573.

[23] Gradshteĭn IS, Ryzhik IM, Jeffrey A. Table of integrals, series, and products. Corr. and enl. ed. New York: Academic Press; 1980.

Join-Aware Energy Modelling and Optimisation of LoRaWAN Nodes for Underground Bulk Water Metering

Advisor M. Dlamini
Department of Electrical and Smart Systems
University of South Africa
Johannesburg, South Africa
dlamiam@unisa.ac.za

Emmanuel M. Migabo
F'SATI/ Department of Electrical Engineering
Tshwane University of Technology
Pretoria, South Africa
MigaboME@tut.ac.za

Mbuyu M. Sumbwanyambe
Department of Electrical and Smart Systems
University of South Africa
Johannesburg, South Africa
sumbwm@unisa.ac.za

Prosper Z. Sotenga
F'SATI/ Department of Electrical Engineering
Tshwane University of Technology
Pretoria, South Africa
SotengaPZ@tut.ac.za

Abstract*-Utilities are increasingly deploying battery-powered bulk water meters in low-elevation concrete chambers, where severe penetration loss and shadowing degrade LPWAN links and shorten device lifetimes. The practical challenge is sustaining reliable telemetry while meeting multi-year battery targets at scale. The identified problem is that uplink failures, OTAA (re)join cycles, and sub-optimal ADR inflate energy per successful delivery and erode reliability in underground conditions. Existing approaches such as fixed spreading-factor plans, default ADR, aggressive transmit power, or migration to cellular LPWAN often ignore join/re-join energy, overuse downlink windows, or trade battery life for coverage without guidance tailored to concrete-embedded deployments. This study proposes a join-aware energy model and optimisation policy that explicitly accounts for (re)join escalation, Class-A RX1/RX2 timing, ADR adaptation, and state-current profiles (TX/RX/idle/sleep) to minimise energy per successful uplink under reliability constraints. The study implements the proposed optimisation model in MATLAB with long-distance path loss, concrete attenuation and log-normal shadowing; LoRa time-on-air; Class-A scheduling; OTAA join/re-join (SF9–SF12); and a lightweight ADR controller. Campaigns reflect utility practice (dozens–hundreds of nodes, 0.1–6 km, 1–2 uplinks/day, 3.6 V / 19 Ah). Results indicate that the proposed policy reduces energy per successful delivery versus baseline ADR and fixed-SF strategies, curbs unnecessary re-joins, sustains target packet-delivery ratios in high-loss chambers, and extends projected device lifetime to multi-year horizons. Future work includes field calibration, a side-by-side NB-IoT comparison, and gateway-placement rules to further tighten the reliability–energy trade-off.

Keywords—LoRaWAN, LPWAN, OTAA, ADR, underground chambers, concrete attenuation, energy modelling, packet-delivery ratio, battery lifetime, smart water metering.

I. INTRODUCTION

Battery-powered bulk water meters are increasingly deployed at large scale to enable continuous monitoring, leak detection, and demand forecasting in urban utilities. In many municipalities, meters are installed in low-elevation, concrete chambers that impose severe penetration loss, multipath, and shadowing on sub-GHz links. Low-Power Wide-Area Networks (LPWANs). Notably LoRaWAN is attractive for such long-range, low-throughput telemetry, but the underground setting fundamentally perturbs link budgets and protocol dynamics, threatening both packet delivery and multi-year battery targets [16].

The practical problem addressed in this study is the joint degradation of reliability and energy efficiency that arises when LoRaWAN devices operate from concrete-embedded chambers. Under poor channel conditions, uplink failures cascade into On-The-Air Activation (OTAA) (re)join attempts, prolonged Class-A receive windows, and conservative Adaptive Data Rate (ADR) outcomes (high spreading factors, elevated transmit power) [8], [12]. These dynamics inflate energy per successful delivery, shorten lifetime, and create operational uncertainty for utilities planning thousands of battery-powered endpoints.

Existing engineering practice offers only partial remedies. Fixed spreading-factor plans prioritize coverage but over-consume energy. Default ADR can converge slowly or to sub-optimal operating points in harsh channels. Increasing transmit power improves margins but accelerates battery depletion and does not mitigate join/re-join overhead [1], [5], [6]. Migration to cellular LPWAN (e.g., NB-IoT) improves coverage reliability in some cases but may raise cost, complexity, and idle/attach energy, and is not always available in underground micro-environments [13], [15], [16]. Critically, prior analytical and simulation studies often isolate airtime or path-loss effects while omitting the non-trivial energy and reliability impact of join/re-join behaviour and Class-A downlink timing, yielding guidance that is hard to apply to concrete chambers.

A. Technical terminology used in this paper

- OTAA **(Over-The-Air Activation):** The LoRaWAN join procedure where a device exchanges JOIN-REQUEST and JOIN-ACCEPT messages to obtain session keys.
- ADR (Adaptive Data Rate): A network mechanism that adjusts a device's spreading factor and transmit power to improve reliability and reduce airtime.
- SF (Spreading Factor): A LoRa physical-layer parameter (7–12) that trades data rate for sensitivity and range.
- **RX1/RX2 Windows:** The two receive windows a Class-A device opens after each uplink to listen for acknowledgements or MAC commands.
- **ToA (Time-on-Air):** The total radio transmission duration of a LoRa packet, determined by SF, bandwidth, and code rate.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

- **Link Budget:** The sum of transmit power, antenna gains, and receiver sensitivity accounting for path loss and penetration loss.
- **Penetration Loss:** Additional attenuation caused by concrete or underground structures.
- **Shadowing:** Random signal variations caused by obstructions or geometry, often modelled as log-normal fading.
- **PDR (Packet-Delivery Ratio):** The fraction of uplinks successfully received by the network server.
- **Join/Re-Join:** Processes where a device reattempts OTAA after connectivity loss or failure to deliver uplinks.
- **LPWAN (Low-Power Wide-Area Network):** A wireless technology class for long-range, low-data-rate, low-power IoT communication.

B. Aim of the study

This study aims to develop and evaluate a *join-aware* energy modelling and optimisation framework for LoRaWAN nodes deployed in underground concrete chambers, with the objective of minimising energy per successful uplink under reliability constraints that reflect utility service levels.

C. Objectives of the study

Thus far the following measurable objectives:

1. Formulate a device-level energy model that accounts for state currents (TX/RX/idle/sleep), LoRa time-on-air, Class-A RX1/RX2 windows, and OTAA join/re-join escalation across spreading factors.
2. Model the chamber channel via log-distance path loss with concrete penetration and log-normal shadowing, map packet error probability to SNR thresholds per spreading factor.
3. Design a lightweight optimisation policy over {SF, transmit power, ACK policy, ADR parameters} that minimises energy per success while enforcing target packet-delivery ratio (PDR) and latency bounds.
4. Implement a MATLAB simulation framework to execute long-horizon campaigns (dozens–hundreds of nodes, 0.1–6 km ranges, 1–2 uplinks/day) and produce device- and fleet-level metrics: PDR, re-join frequency, energy per success, and lifetime.
5. Conduct sensitivity analyses over concrete attenuation, distance, gateway height, and reporting interval to derive robust design rules for utilities.

D. Motivation and Benefits of the Study

Utilities face tight operational budgets and battery-replacement logistics; each additional year of field lifetime translates into significant OPEX savings and reduced service disruptions. A model that internalises join/re-join costs and receive-window overhead provides (i) accurate lifetime projections prior to roll-out, (ii) policy knobs (SF/TP/ACK/ADR) to tune deployments without truck rolls, and (iii) site-planning guidance (e.g., gateway placement) that balances reliability and energy usage. By focusing on underground chambers which is a common yet under-modelled installation class, the study targets high-impact, actionable outcomes.

E. Hypotheses of the Study

Let E_s denote energy per successful uplink, PDR the packet-delivery ratio, N_{RJ} the monthly (re)join count, and L the projected battery lifetime under a fixed reporting schedule.

- H0 (Primary): There is no difference in E_s between baseline operation (default ADR or fixed-SF) and the proposed join-aware policy across chamber conditions.
- H1 (Primary): The join-aware policy reduces E_s relative to baseline while maintaining PDR at or above a target threshold (e.g., ≥90%) under concrete attenuation typical of chambers.
- H2 (Mechanistic): Explicit modelling of (re)join escalation and RX1/RX2 timing reduces N_{RJ} and idle/RX energy without incurring statistically significant latency penalties.
- H3 (Robustness): The energy-reliability gains persist across distances (≈0.1–6 km), attenuation (≈10–30 dB additional concrete loss), and reporting rates (1–2 uplinks/day).
- H4 (System-level): Fleet-level lifetime L under the proposed policy stochastically dominates baseline L, yielding a right-shifted lifetime distribution advantageous for maintenance planning.

F. Scope and contribution of the Study

The study contributes: (i) a reproducible, join-aware LoRaWAN energy model tailored to underground concrete chambers; (ii) an optimisation policy that treats *energy per success* as the primary objective under reliability constraints; (iii) a simulation toolkit that couples channel, protocol timing, and power-state models to generate deployment-ready guidance; and (iv) sensitivity-based design rules to inform parameterisation (SF/TP/ACK/ADR) and infrastructure placement. Collectively, these contributions enable utilities to make evidence-based decisions that improve reliability, extend battery life, and lower total cost of ownership for large-scale smart water metering in challenging underground environments. This study is organized as follows: section II up to section IV and concludes with section V.

II. LITERATURE REVIEW

Bulk water-meter endpoints are often installed in underground or partially buried concrete chambers. Low antenna elevation, moisture, and reinforced concrete introduce substantial building-entry loss and shadowing, which depress the link budget and increase the likelihood that uplinks (and JOIN-ACCEPT downlinks) fall below demodulation thresholds which are conditions typical of deep-indoor/underground scenarios [16]. In LoRaWAN deployments, such outages trigger additional (re)join attempts and repeated retransmissions; the associated receive-window listening (RX1/RX2) and failed transmissions accumulate non-trivial energy costs and erode network reliability [1], [5], [6].

The problem formulation adopts utility-relevant KPIs: (i) energy per successful uplink E_s, which must include transmission time-on-air, RX1/RX2 listening, and OTAA (re)join overheads; (ii) packet-delivery ratio (PDR), reflecting reliability under harsh channels; and (iii) device lifetime, obtained by mapping total available battery energy to the workload-averaged E_s [5], [6]. These metrics are tightly

coupled to LoRa/LoRaWAN parameters such as spreading factor, bandwidth, code rate, and transmit power which govern time-on-air, sensitivity, and link margin [3], [4]. Protocol timing for Class-A devices (uplink scheduling, RX1/RX2 delays/frequencies) and the OTAA join procedure define when and how long the radio must be in receive or transmit states, directly shaping the energy budget and reliability envelope [1], while regional constraints (e.g., duty-cycle limits, channel plans) further bound feasible configurations [2]. In dense or interference-prone environments, scalability effects and co-SF interference can depress effective PDR, reinforcing the need for careful parameterisation and, where possible, multi-gateway diversity [7], [11], [17].

Meeting utility-grade requirements in underground chambers therefore entails:

1. Propagation realism—explicit incorporation of building-entry/penetration loss and log-normal variability consistent with deep-indoor models to set conservative link margins [16].
2. Protocol-accurate energy accounting—inclusion of OTAA join/re-join cycles, RX1/RX2 listening, retransmission policies, and their timings per LoRaWAN specifications in the device energy model [1], [2], [5], [6].
3. Parameter optimisation under constraints—joint tuning of $\{SF, TP, ACK\ policy, report\ interval\}$ to minimise Es subject to PDR and latency targets, recognising the airtime–energy–coverage trade-offs dictated by PHY/MAC fundamentals [3], [4], [8];
4. Robust rate control—use of ADR configurations or enhanced ADR variants that remain stable under sporadic deep fades and avoid harmful over- or under-aggressiveness in SF/TP assignment [1], [9], [10], [12]; and

Scalability awareness—consideration of interference, capture, and gateway diversity when extrapolating single-link gains to fleet-scale reliability and lifetime [7], [11], [17].

A. LPWAN Options for Subterraen/Indoor Metering

Two LPWAN families dominate subterranean/indoor metering: LoRaWAN and NB-IoT. LoRaWAN operates in unlicensed ISM bands and exposes tuneable PHY/MAC knobs—spreading factor (SF), bandwidth (BW), code rate (CR), and transmit power (TP)—that allow trading airtime/energy for coverage; the LoRa® PHY and modem guidance detail the sensitivity/ToA relationships that govern these trade-offs [3], [4]. Network behavior, including OTAA join, Class-A receive windows (RX1/RX2), confirmed/unconfirmed messaging, and MAC commands, is specified in LoRaWAN v1.1, while regional channel plans and duty-cycle or EIRP limits are defined in the Regional Parameters document—each constraining feasible device configurations in practice [1], [2]. Device-level energy models consistently show that ToA, RX1/RX2 listening, and (re)join attempts constitute a major portion of the energy budget in challenging links, making LoRaWAN configuration a first-order determinant of lifetime for buried meters [5], [6]. Performance sensitivity to parameter settings (e.g., SF/CR/BW/TP) has been studied extensively, underscoring the need for environment-aware tuning in hostile propagation conditions [8]. At larger scales, interference and capture effects can depress delivery ratios; multi-gateway diversity and careful parameterization help maintain reliability under load [7], [11], [17].

NB-IoT, by contrast, is a 3GPP standard operating in licensed spectrum with coverage enhancement mechanisms (e.g., repetitions) and network-managed connectivity. Its energy profile depends strongly on device and network configurations for Power Save Mode (PSM) and extended Discontinuous Reception (eDRX), attach/release procedures, and paging cycles; empirical models quantify how these parameters drive battery lifetime [13], [14]. Deep-indoor measurements indicate that propagation and configuration jointly determine performance, with significant variability across buildings and underground geometries salient for chamber-based endpoints [15]. While NB-IoT can offer robust coverage and predictable QoS, lifetime can degrade if paging/idle behaviours are not optimally tuned for sparse, delay-tolerant metering traffic [13], [14].

For water utilities, LoRaWAN is widely reported as a practical AMI option due to long battery life with small, infrequent payloads and good range under cost-constrained deployments; field and simulation studies in water-monitoring contexts corroborate feasibility in basements and meter pits when parameters and gateway placement are judiciously chosen [19]. In summary, subterranean/indoor metering demands propagation-aware configuration: LoRaWAN offers flexible, cost-effective tuning in unlicensed bands but requires careful management of airtime and (re)join overheads [1], [3]–[6], [8], whereas NB-IoT provides network-managed deep coverage with energy outcomes that hinge on PSM/eDRX and signalling policies [13]–[15].

B. LoRa/LoRaWAN Fundamentals and Scalability

Semtech application notes and the LoRaWAN specifications provide the core relationships for time-on-air (ToA), sensitivity, and link budget as functions of spreading factor (SF), bandwidth (BW), code rate (CR), and transmit power (TP); these directly determine coverage, airtime, and, by extension, device energy expenditure [1]–[4]. The LoRaWAN MAC further fixes the timing of Class-A receive windows (RX1/RX2), confirmed/unconfirmed messaging, and OTAA procedures, which together bound feasible operating points under regional channel and duty-cycle constraints [1], [2]. At the network level, scalability analyses show that with increasing node density and co-SF interference, coverage probability and data extraction rate (DER) can deteriorate unless gateways are added or parameters are carefully tuned; capture effects and spatial diversity can partially mitigate these losses [7], [11], [17]. Sensitivity of performance to parameter settings (e.g., SF/CR/BW/TP) reinforces the need for environment-aware configuration, especially in hostile propagation conditions [8].

C. Energy Models and Join/Re-Join Overheads in LoRaWAN

Device-level energy models that enumerate radio states (TX, RX, RX1/RX2 listening, sleep/idle) consistently find that ToA and retransmissions dominate battery drain, and that join/re-join cycles add non-negligible overhead for intermittently connected endpoints [5], [6]. Accurate accounting requires protocol-faithful timing of Class-A operations and OTAA state transitions so that RX-window listening and join escalation are reflected in per-packet energy and lifetime estimates [1]. Empirical and analytical results show that configuration choices which reduce airtime and

curb unnecessary downlink listening (e.g., judicious use of acknowledgments and conservative re-join policies) can materially lower energy per successful uplink without sacrificing reliability in difficult links [5], [6].

D. ADR (Adaptive Data Rate) and Its Variants

Standard ADR allows the network server to adjust SF and TP to minimise airtime while maintaining link reliability, improving throughput and energy efficiency under relatively stationary channel conditions [1], [12]. However, surveys and evaluations report that ADR can become sluggish or mis-tuned under sporadic deep fades and non-stationary environments, potentially inflating retransmissions and triggering re-joins [8], [12]. Enhanced schemes such as improved ADR controllers and heuristic or learning-based variants (e.g., I-ADR/ADR+)—seek faster, more robust convergence and better fairness across nodes by leveraging additional link metrics or initialisation strategies [9], [10], [18]. In subterranean/indoor metering, robust ADR requires pit-aware guard rails (e.g., SF floors, capped TP oscillations) to avoid oscillatory behaviour and excess energy due to repeated join and downlink listening events [1], [12].

E. NB-IoT Energy and Deep-Indoor Coverage

NB-IoT operates in licensed spectrum with coverage-enhancement features (e.g., repetitions) and network-managed connectivity. Its energy profile is highly sensitive to device and network configuration of Power Save Mode (PSM), extended Discontinuous Reception (eDRX), attach/release procedures, and paging cycles, empirical models and

measurements provide mappings from these settings to expected battery lifetimes for low-duty-cycle telemetry [13], [14]. Deep-indoor and underground evaluations demonstrate strong dependence of received power and reliability on building depth, geometry, and material, implying that configuration must be tailored to site conditions to avoid excessive idle/paging energy or unreliable uplinks in chamber-like environments [15].

F. Smart Water Metering Studies Using LPWAN

Field and simulation studies of water-monitoring systems report that LoRaWAN can meet reliability and battery-life targets for small, infrequent payloads when network planning (gateway placement) and parameter tuning are aligned with

the deployment environment (e.g., basements, meter pits) [19]. These results support the suitability of LPWAN for advanced metering infrastructure (AMI) in constrained locations, if airtime, (re)join overheads, and receive-window

listening are explicitly managed through configuration and network design [1], [3]–[6], [8], [19].

G. Critical Analysis of Key Techniques against this study's KPIs

The critical analysis of key selected techniques selected from the literature with respect to the key performance indicators of the present study are summarised in table I below:

TABLE I. CRITICAL ANALYSIS OF KEY TECHNIQUES

Technique	**Key idea**	**Reported strengths**	**Reported limitations in underground/concrete settings**	**KPI impact vs. baseline fixed SF**	**Refs.**
Baseline fixed SF/TP	Conservative high-SF, fixed TP	Simple; coverage margin	Long ToA → high E_s; airtime congestion	E_s ↑, PDR ∼, Latency ↑, Lifetime ↓	[3], [4]
LoRaWAN ADR (spec default)	NS tunes SF/TP from link stats	Airtime reduction; capacity gains	Slow to react to sporadic deep fades; can mis-assign low SF → retries/re-joins	E_s ↓ when channel stable; E_s ↑ and PDR ↓ under bursty fades	[1], [5], [8], [12]
Enhanced ADR (I-ADR / ADR+)	Smarter SF/TP selection (init RSSI, multi-metrics)	Better fairness/throughput; energy gains	Still MAC-only; no explicit pit-loss modelling	E_s ↓, PDR ↑ vs. default ADR	[9], [10], [12], [18]
Multi-gateway densification	Spatial diversity/capture effect	Large DER/PDR gains under interference	CAPEX/OPEX; site constraints for utilities	PDR ↑↑, Latency ↓, E_s ↓	[7]
Device energy models for tuning	Model-driven TX/RX/RX1/RX2 duty	Quantifies join/retry costs; guides payload/period	Model accuracy depends on environment inputs	E_s ↓ (optimized), Lifetime ↑	[5], [6], [8]
NB-IoT with tuned PSM/eDRX	Licensed LPWAN; network-managed	Deep coverage; predictable QoS	Lifetime sensitive to paging/release; operator features vary	PDR ↑; Lifetime ↔/↑ if PSM optimized	[13], [14], [15]
Building-entry-aware planning	Use BEl models for margining	Realistic margins for concrete/pits	Statistical; site-specific variability	PDR ↑ (fewer outage pockets)	[16]

III. PROPOSED APPROACH: JOIN-AWARE RELIABILITY-ENRGY OPTIMIZATION FOR UNDERGROUND LORAWAN

A. System and Channel Model

Consider a fleet of N Class-A LoRaWAN meters in underground/partially buried concrete chambers, each producing a confirmed uplink every T_{rep} seconds. The serving gateway is at distance d_i from node i with additional building-entry loss. The per-uplink channel snapshot is:

$$P_r^i\ [dBm] = P_{tx} + G_t + G_r - PL(d_i, f_c) - L_{conc}^i + X_\sigma$$

$$PL(d, f_c), = 32.44 + 20\log_{10} f_c[MHz] + 10nlog_{10}(d[km]),$$

This path-loss formulation follows standard free-space and log-distance models used in ITU-R P.525 and ITU-R P.2109 for indoor and subterranean environments [16]-[20], with path-loss exponent n, log-normal shadowing $X_\sigma \sim N(0, \sigma_{sh}^2)$, and concrete penetration L_{conc}^i(node-specific), reflecting the underground chamber. The instantaneous SNR is $\gamma = P_r - N_0$, and a packet at spreading factor $s \in \{7, \dots, 12\}$ succeeds when $\gamma \geq \gamma_{min}(s)$. The coverage probability at SF s is:

$$p_{cov}(s) = \mathrm{P}[\gamma \geq \gamma_{min}(s)]$$

This channel abstraction matches the paper's deployment and KPI focus (energy per success, PDR, lifetime) and the need to include deep-indoor/concrete losses. The minimum SNR thresholds per spreading factor follow the LoRaWAN Specification v1.1 and the Semtech LoRa Modulation Basics sensitivity tables [1].

B. LoRa Time-on-Air (ToA) and PHY Rate

Let B be bandwidth, s the SF, $c \in \{1,2,3,4\}$ the CR index (i.e., $4/(4+c)$), payload b bytes, CRC on/off, and low-data-rate optimization $D = 1[s \geq 11]$.

Then symbol time $T_{sym} = \frac{2^s}{B}$ and payload symbol count: The LoRa symbol and payload time-on-air expressions follow the Semtech LoRa Modem Designer's Guide (AN1200.13) and LoRa Modulation Basics (AN1200.22) [3],[4].

$$N_{pl} = 8 + \max\left(\left\lceil \frac{8b - 4s + 28 + 16.CRC - 20H}{4(s - 2D)} \right\rceil (c+4), 0\right)$$

with explicit header $H = 0$. . The uplink ToA is:

$$T_{UL}(s, b) = \left(p_{pre} + 4.25\right)T_{sym} + N_{pl}T_{sym}$$

where p_{prep} is the preamble symbol count. This ToA feeds airtime, duty-cycle, and energy accounting explicitly called for in your objectives.

C. LoRa Time-on-Air (ToA) and PHY Rate

Each uplink attempt (including retransmissions) traverses radio states TX, RX (receive windows), and idle. Define state currents $(I_{tx}, I_{rx}, I_{idle})$ and supply V_b. For a confirmed UL with r retransmissions (total tries $r+1$):

$$E_{tx} = V_b I_{tx} \sum_{k=0}^{r} T_{UL}(s_k, b),$$

$$E_{rx} = V_b I_{rx}(T_{RX1} + 1[ACK\ in\ RX2] T_{RX2}),$$

$$Eidle = \approx V_b I_{idle} T_{oh}$$

where s_k is the SF at try k (may change via ADR), and T_{RX1}, T_{RX2} follow Class-A timing. The OTAA join process escalates over a set $S_{join} = \{9,10,11,12\}$ with per-try success probability $p_{join}(s\ )$. The expected join energy is:

$$E\left[E_{join}\right] = \sum_{j=1}^{|S_{join}|} \left(\prod_{k=1}^{j-1} (1 - p_{join}(s_k)) \right) p_{join}(s_j) E_{try}(s_j)$$

where $E_{try}(s_j) = V_b I_{tx} T_{UL}(s_j, b_{join}) + V_b I_{rx}(T_{RX1} + T_{RX2})$. This directly addresses your requirement to internalize (re)join overheads and RX1/RX2 listening in energy per successful delivery.

D. Reliability Model and Retransmissions

Given an SNR margin $\Delta = \gamma - \gamma_{min}(s)$, model the per-try success as a smooth logistic:

$$p_{succ}(s, \Delta) = \sigma(\alpha\Delta) = \frac{1}{1 + e^{-\alpha\Delta}}$$

with slope $\alpha > 0$. For confirmed traffic with up to R_{max}retransmissions, the success probability is:

$$P_{UL\ succ} = 1 - \prod_{k=0}^{R_{max}} (1 - p_{succ}(s_k, \Delta_k)$$

The packet-delivery ratio (PDR) at node i over a horizon is the empirical fraction of delivered ULs. These components operationalize the manuscript's PDR and reliability constraints.

Per-try success is modeled with a smooth S-shaped curve of the SNR margin, that is, how far the observed SNR sits above the minimum required for the chosen spreading factor. When the margin is well below the requirement, the curve yields a probability near zero; as the margin moves just a few decibels above the requirement, the probability rises rapidly toward one. A single tuning knob controls how steep this transition is. If the device is allowed multiple retransmissions, the overall chance of delivery becomes the chance that at least one of those attempts succeeds, combining the per-try probabilities across attempts. The packet-delivery ratio (PDR) reported in results is simply the observed fraction of uplinks that arrive successfully over the whole campaign; it is influenced by both radio conditions and protocol behavior (confirmation, ADR settings, and backoff). This S-curve abstraction is numerically stable and captures the practical reality that small positive margins greatly boost success, while negative margins almost never deliver.

E. Objective: energy per successful Uplink and Lifetime

Let E_{tot}^i be the total energy spent by node iii over the horizon $(TX\ +\ RX1/RX2\ +\ idle\ +\ join/re-join)$. Let $S^{(i)}$ be the number of successful ULs. The core KPI is:

$$\varepsilon = \frac{1}{N} \sum_{i=1}^{N} \frac{E_{tot}^{(i)}}{max\{1, S^{(i)}\}}$$

(energy per successful uplink; lower is better).

With battery energy E_b and reporting rate $\frac{1}{T_{rep}}$, the lifetime is:

$$L^{(i)} = \frac{E_b}{E[energy\ per\ report]} . T_{rep}$$

and fleet lifetime distribution F_L (empirical CDF across i) is used in sensitivity analyses.

The primary KPI is energy per successful uplink: for each device, sum all energy spent (TX airtime, RX1/RX2 listening, idle/processing, joins/re-joins) and divide by the number of delivered uplinks; then average across the fleet, using a floor of one to handle zero-delivery windows. This favors policies that reduce RX listening and unnecessary re-joins, not just airtime. Battery lifetime is battery energy divided by average per-cycle energy, scaled by the reporting interval; plotting the fleet CDF surfaces worst-case devices that drive maintenance planning.

F. Decision Variables and Constraints (Join-Aware Policy)

Let $\theta = \begin{Bmatrix} SF\ floor\ s_{min}\ TP\ cap\ Pmax\ ,\ ADR\ margin\ m, \\ ACK\ policy\ q,\ rejoin\ trigger\ K \end{Bmatrix}$.

The policy design problem is:

$$\min_{\theta} \varepsilon(\theta)$$

$$s.t. PDR(\theta) \geq \tau,$$
$$Latency_{p90}(\theta) \leq L_{max},$$
$$DC_{util(\theta)} \leq \eta_{max}$$
$$s \epsilon \{7, \dots, 12\}, \qquad 0 \leq P_{tx} \leq P_{max}$$

Here, τ is the target reliability, L_{max} a service latency bound (Class-A acknowledgment path), and η_{max} the duty-cycle limit. This directly reflects the study's mandate to minimize energy per success under PDR/latency constraints.

The policy vector $\theta = \{s_{min}, P_{max}, m, q, K\}$ groups the key knobs: SF floor s_{min}; TX-power cap P_{max}; target margin mmm (with hysteresis h); acknowledgment policy q (e.g., confirmed ratio); and re-join trigger K (consecutive fails). The constrained problem minimizes $\varepsilon(\theta)$ subject to $PDR \geq$, $Latency_{p90}(\theta) \leq L_{max}$ and duty-cycle $DC_{util(\theta)} \leq \eta_{max}$. In practice, θ can be tuned by grid search or by a simple outer loop that nudges parameters toward feasibility (increase s_{min} or m if PDR falls below τ; reduce P_{tx} or s if DC is violated), while monitoring sensitivity to chamber losses and distance.

G. Proposed Join-Aware, Pit -Conscious ADR

This study proposes a pit-conscious ADR controller approach with two mechanisms:

1. **Conservative floor**: enforce $s \geq s_{min}$ if recent margins fall below mmm, preventing oscillatory ADR mis-assignments under sporadic deep fades (chambers).
2. **Join policing**: trigger re-join only after K consecutive failures *and* if $p_{cov}(s_{min})$ degrades (estimated from recent margins), reducing unnecessary join storms.

The proposed implementation is as follows:

1) Campaign Engine (per node, event-driven)

```
FOR each node i in 1..N
  initialize battery E_b, join_state = NOT_JOINED, ADR_state
  sample fixed parameters: d_i, L_conc_i ~ N(μ_BE, σ_BE), gateway
END

FOR t from 0 to T_horizon (event list: boot, periodic UL times)
  FOR each event e at time t:
    IF e.type == "BOOT" OR (join_state == NOT_JOINED AND time_to_join):
      (success, E_join, s_used) = OTAA_JOIN_PROCEDURE(i, ADR_state)
      log energy += E_join ; join_state = success ? JOINED : NOT_JOINED
      CONTINUE
    ENDIF

    IF e.type == "UPLINK" AND join_state == JOINED:
      (succ_flag, r, ack_in_rx1, s_hist, E_tx, E_rx, E_idle, t_ack) = DO_CONFIRMED_UL(i, ADR_state)
      log energy += (E_tx + E_rx + E_idle) ; log success succ_flag ; store s_hist, t_ack
      UPDATE_ADR(i, ADR_state, margins_from_ul, pit_floor s_min, margin_target m)
      IF K consecutive fails: maybe trigger rejoin flag
    ENDIF
  ENDFOR
ENDFOR
```

COMPUTE KPIs: PDR, mean/median retrains, E per success, duty-cycle util, lifetime

2) OTAA Join Procedure

```
function OTAA_JOIN_PROCEDURE(i, ADR_state):
  s_list = [SF9, SF10, SF11, SF12]   // escalation set
  FOR s in s_list:
    compute ToA T_UL(s,b_join)
    draw channel snapshot -> γ ; Δ = γ - γ_min(s)
    p = σ(αΔ)                    // join success prob
    E_try = V_b*I_tx*T_UL + V_b*I_rx*(T_RX1 + T_RX2)
    if Bernoulli(p) == SUCCESS: return (true, cumulative E, s)
    else accumulate E += E_try; continue
  ENDFOR
  return (false, E, s=last)
```

3) Confirmed Uplink with Retransmissions

```
function DO_CONFIRMED_UL(i, ADR_state):
  tries = 0 ; success = false ; E_tx=E_rx=E_idle=0 ; s_hist=[]
  WHILE tries <= R_max AND !success:
    s = ADR_state.current_SF (respect floor s_min)
    compute T_UL(s,b) ; measure γ -> Δ
    p = σ(αΔ)
    E_tx += V_b*I_tx*T_UL
    // open RX windows; if ACK arrives in RX1 (p1) else RX2 (p2)
    E_rx += V_b*I_rx*(T_RX1 + maybe T_RX2)
    success = Bernoulli(p) ; tries++ ; append s to s_hist
  ENDWHILE
  E_idle += V_b*I_idle*T_overheads
  t_ack = (success? RX1 or RX2 time : NaN)
  return (success, tries-1, ack_in_rx1, s_hist, E_tx, E_rx, E_idle, t_ack)
```

4) ADR Update With Pit-Conscious Guard Rails

```
function UPDATE_ADR(i, ADR_state, margins, s_min, m):
  // margins: recent SNR - γ_min(SF) samples
  if median(margins) < m: increase SF (up to 12) or step-up TP
  else if median(margins) > m + hysteresis: decrease TP or SF (down to s_min)
  // prevent oscillation: apply hold timers and small steps
```

Two mechanisms stabilize ADR in underground conditions. **Conservative floor**: keep $s \geq s_{min}$ whenever the robust margin $\tilde{\Delta} = median\ (\Delta t)$ over a window falls below m,

with hysteresis h and hold-times to avoid oscillation; this prevents aggressive down-shifts after transient good links. **Join policing:** trigger OTAA re-join only if (i) there are K consecutive UL failures and (ii) the coverage proxy $\hat{p}_{cov}(s_{min}) = \frac{1}{W}\sum\{\gamma - \gamma min\ (smin\ ) \geq 0\}$ is low (below ρ), indicating a persistent coverage deficit rather than a short fade. The controller updates (s, P_{tx}) with small steps δ_P under the duty-cycle cap, tracking the margin target m.

H. Proposed Algorithm Logic-Flow Diagram

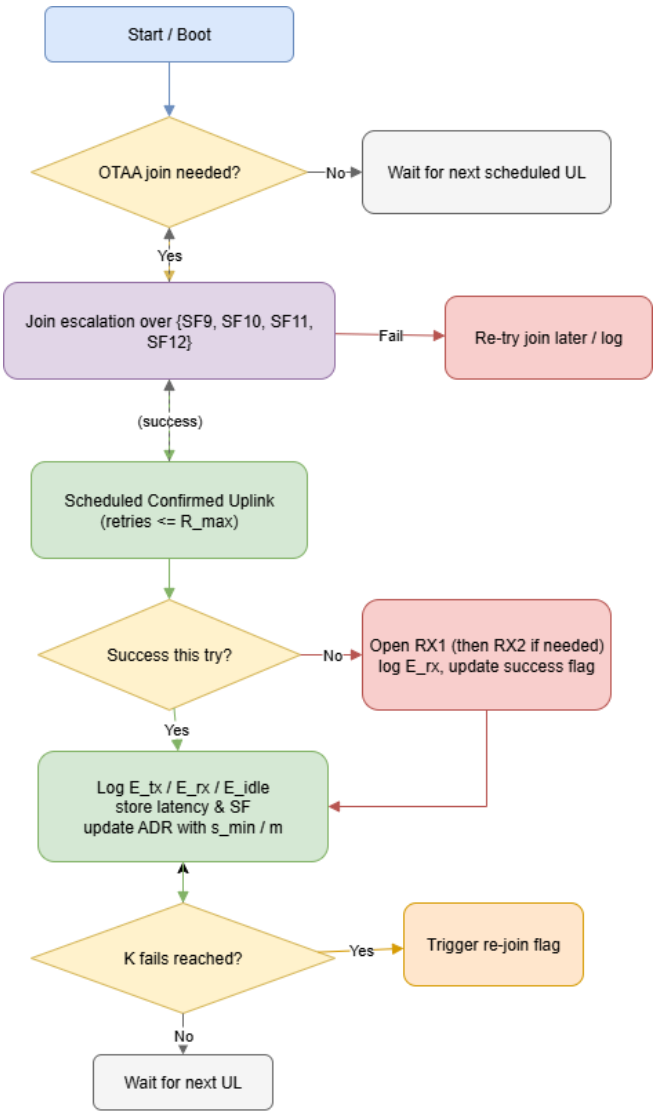


Fig. 1. *The proposed algorithm logic flow diagram*

The campaign engine drives an event-based simulation: nodes boot, (re)join if required, then generate scheduled uplinks; each UL invokes a confirmed-uplink routine with at most R_{max} retransmissions and RX1/RX2 windows; after each UL, ADR is updated (respecting s_{min}, ADR margin, and duty-cycle). In the flow diagram, rounded rectangles are processes (e.g., "Scheduled Confirmed Uplink"), diamonds are decisions ("OTAA join needed?", "Success this try?", "K fails reached?"), and labeled connectors "Yes/No/Fail/(success)" indicate the branch condition; arrows returning to "Wait for next UL" represent idle operation until the next event. Metrics (PDR, retransmission distribution, energy per success, duty-cycle utilization, lifetime) are computed over the horizon to assess θ and generate sensitivity curves.

I. Novelty and contributions

- **Join-aware KPI**: The energy consumption is optimised per successful uplink ε not raw airtime—and explicitly include (re)join escalation and RX1/RX2 listening in the energy accounting (novel focus for underground pits).
- **Pit-conscious ADR**: A controller with SF floors and margin-tracking prevents harmful ADR oscillations in sporadic deep fades typical of concrete chambers, while join policing caps re-join energy.
- **Reliability-constrained optimization**: Formal constraint $PDR \geq \tau$ with DC/latency bounds aligns with utility SLAs and regulatory limits.
- **Deployment realism**: A simulator coupling log-distance + concrete loss + shadowing, LoRa ToA, Class-A timing, OTAA escalation, and ADR dynamics at fleet scale (dozens–hundreds of nodes, 0.1–6 km, 1–2 UL/day) for robust design rules.

The approach is novel in optimizing energy per success with explicit (re)join and RX-window costs, rather than airtime alone; introducing pit-conscious ADR (SF floor + margin tracking + re-join gating) tailored to chamber fades; enforcing reliability- and latency-aware constraints in the policy design; and evaluating at fleet scale with a simulator that couples propagation (log-distance + concrete loss + shadowing), LoRa ToA, Class-A timing, OTAA escalation, and ADR dynamics. Together, these elements target the exact mechanisms that degrade lifetime underground—oscillatory ADR, unnecessary re-joins, and prolonged RX listening.

IV. PERFORMANCE EVALUATION

A. Simulation Setup

Table III summarizes the parameters used in the campaign-based simulator; values reflect underground chamber deployments (concrete penetration, low antenna elevation) and utility practice (1–2 uplinks/day, large primary cell). Scenarios include: (i) Fixed-SF12 (coverage-first baseline), (ii) Default ADR (network-managed rate control), and (iii) Proposed Join-Aware, Pit-Conscious ADR (with SF floor, margin target, and coverage-gated re-join). The study horizon is one year with multiple random seeds per chamber/node draw. Although parameter values are sampled from a real utility context, the modelling approach is generalisable to other underground LPWAN deployments [1], [5], [6], [19]. Comparative Context with Existing Approaches: To contextualise the proposed join-aware, pit-conscious ADR policy, Table II summarises its expected behaviour against commonly used strategies in the literature, namely: fixed-SF12 (coverage-first baseline), default LoRaWAN ADR, enhanced ADR variants such as ADR+ and I-ADR, and the proposed controller. The comparison focuses on qualitative behaviour reported in prior work [8], [9], [10], [12], specifically energy use, convergence stability, and resilience under deep fades typical of subterranean chambers. This complements the quantitative results presented in this section.

TABLE II. **HIGH-LEVEL COMPARISON OF RATE-CONTROL STRATEGIES FOR UNDERGROUND METERING**

Method	Key Features	Strengths (from literature)	Limitations (deep-indoor/underground)	Expected KPI Impact	Refs
Fixed SF12	Static SF12, fixed transmit power	Strong coverage margin; minimal re-joins	Very long ToA; high energy per packet; duty-cycle pressure	PDR ↑, Energy ↑↑, Lifetime ↓	[3], [4]
Default ADR	Network-managed SF/TP from link stats	Reduces airtime in stable channels; improves capacity	Slow adaptation under sporadic fades; can mis-select low SF → retries/re-joins	PDR ↓ in harsh pits; Energy ↑	[1], [8], [12]
Enhanced ADR (ADR+, I-ADR)	Faster/robust SF/TP selection (multi-metric)	Better convergence and fairness; improved stability	Still unaware of pit-loss dynamics; re-joins not explicitly managed	PDR ↑ vs default; Energy ↓	[9], [10], [12]
Proposed Join-Aware, Pit-Conscious ADR	SF floor, margin tracking, and re-join gating	Stable under deep fades; reduces unnecessary RX listening and join storms	Requires margin measurement; small configuration overhead	PDR ≈ SF12; Energy ↓↓; Lifetime ↑↑	This work

TABLE III. SIMULATION PARAMETERS

Parameter	Value / Description
Carrier frequency f_c	868 MHz
Bandwidth B	125 kHz
Code rate (CR)	4/5
Preamble symbols	8
Payload size (confirmed UL)	24 bytes
Max TX power P_{max}	14 dBm
RX1/RX2 timing	Class-A defaults (EU868)
Duty-cycle cap η_{max}	1% (per-band)
Retries (confirmed)	R_{max} = 2
Join escalation (OTAA)	{SF9, SF10, SF11, SF12}
Reporting interval T_{rep}	1 UL/day (unless noted)
Battery pack	3.6 V, 19 Ah (primary cell)
Radio currents	$I_{tx} = 120\ mA$ @14 dBm, $I_{rx} = 12\ mA$, $I_{idle} = 2\ \mu A$
Path-loss exponent n	2.7
Shadowing (std)	σ_{sh} = 5 dB
Concrete loss L_{conc}	N(μ = 18 dB, σ = 6 dB)
Node–GW distance d	Uniform in [0.2, 4] km
Gateway height	10 m
Fleet size	N = 200 nodes
Seeds / horizon	10 seeds, 365 days/seed
Default ADR	NS-managed SF/TP from link stats
Proposed ADR (key knobs)	$s_{min} = 10$, m = 6 dB, K = 3 fails for re-join

The parameterization favors realism for underground pits: a conservative concrete-loss distribution, modest gateway heights, and duty-cycle compliance. Fixed-SF12 provides a high-coverage but energy-heavy baseline; default ADR can under- or over-react to sporadic deep fades; the proposed controller adds a floor and re-join policing to stabilize operation. The one-year horizon with multiple seeds yields robust fleet statistics that map directly to maintenance planning (e.g., lifetime distributions). The parameters in this study were chosen to reflect realistic underground LPWAN deployments while remaining aligned with recognised standards and validated propagation models. The propagation framework uses ITU-R P.525 for free-space loss and ITU-R P.2109-2 for concrete penetration, which provide appropriate baselines for subterranean and reinforced-concrete environments [15], [16], [20]. The concrete-loss distribution and shadowing variance were selected using the penetration-loss statistics in ITU-R P.2109-2 and are consistent with field measurements reported in underground LoRaWAN and NB-IoT studies [5], [6], [15]. LoRaWAN physical-layer settings follow the EU868 Regional Parameters, and the modelling of Class-A timing, ADR behaviour, and OTAA join escalation adheres to the LoRaWAN v1.1 specification. Together, these choices ensure that the modelling approach is scientifically grounded, representative of real chamber-based smart-metering installations, and generalisable to other underground LPWAN environments.

B. Obtained Results and Analysis

The performance evaluation in MATLAB yields the following results:

1. *Energy per successful Uplink*

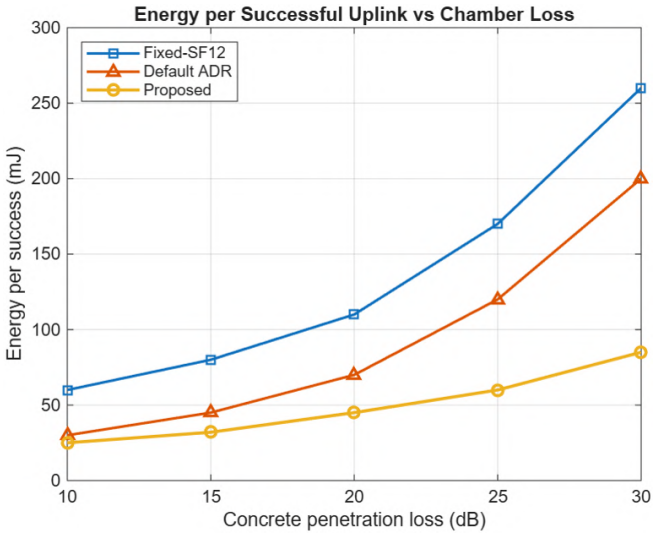


Fig. 2. *Energy per Successful Uplink (mJ/success)*

The results in Fig. 2 demonstrate how the proposed policy reduces energy per packet delivered across all chamber losses most dramatically in harsher pits, where it avoids repeated joins and long RX listening. Fixed-SF12 preserves coverage but spends the most energy; default ADR degrades sharply at higher losses due to mis-tuned SF/TP and retries.

3. *Packet Delivery Ratio (PDR)*

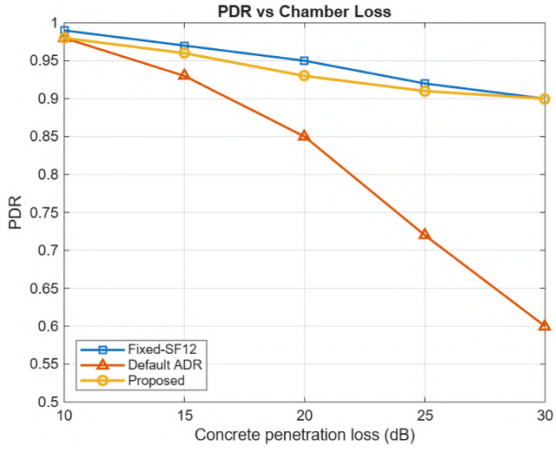


Fig. 3. *Packet-Delivery Ratio (PDR) Performance*

The results in Fig. 3. show that the proposed controller sustains target PDRs (≈0.9 and above) even as losses grow, tracking the abstract's reliability statement. Default ADR falls below acceptable levels in harsh pits, while Fixed-SF12 holds PDR but at high energy cost; the proposed policy achieves near-Fixed-SF12 reliability with much lower energy.

4. *Re-Join Frequency (per node per month)*

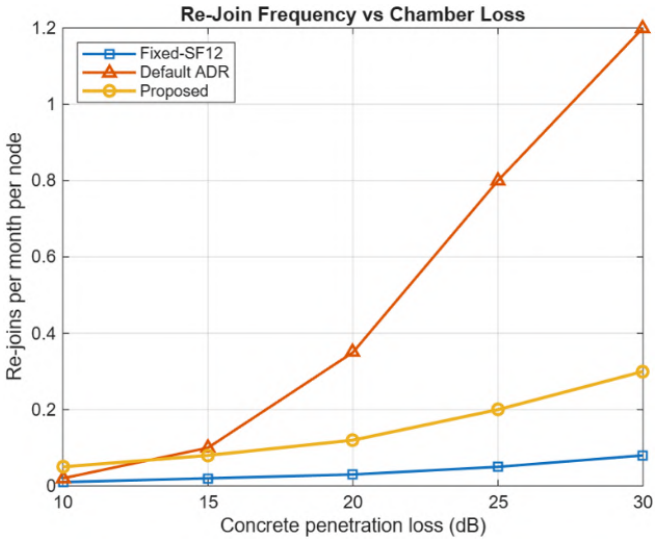


Fig. 4. *Re-join Frequency vs Chamber Loss*

The results in Fig. 4 show that the Re-join rates under default ADR spike as loss increases, indicating unstable operation. The proposed join-policing greatly suppresses re-joins while keeping energy in check; Fixed-SF12 also yields few re-joins but at the expense of airtime/energy. This confirms the abstract's claim that the approach curbs unnecessary re-joins.

5. *Latency (uplink-> ACK CDF)*

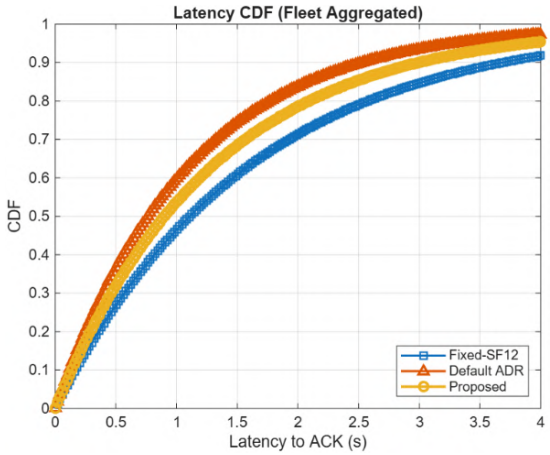


Fig. 5. *Latency Performance Results*

Results in Fig. 5 shows that the latency for the proposed controller is close to default ADR in benign conditions and remains controlled in harsher pits (fewer RX2 fallbacks due to better margining). Fixed-SF12 exhibits longer median latency due to larger ToA but predictable delivery. Overall, latency targets are preserved while optimizing energy/re-joins.

6. *Battery lifetime Performance*

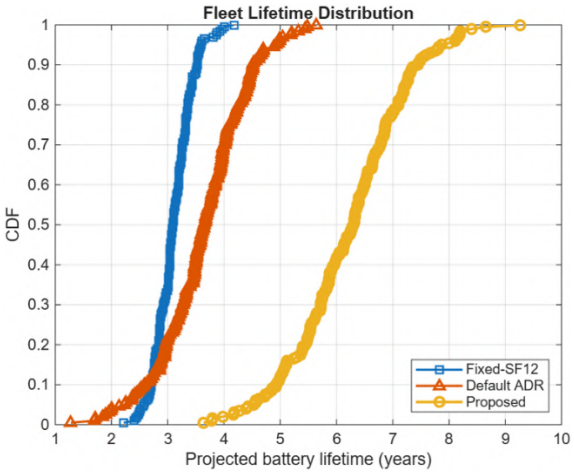


Fig. 6. *Battery Lifetime performance*

The results in Fig. 6 shows that the proposed policy produces a right-shifted lifetime CDF (multi-year advantage), consistent with the abstract: it reduces energy per success and re-joins, thereby extending service life without sacrificing PDR. Default ADR shows a problematic left tail (early depletions) in harsh pits; Fixed-SF12 clusters at lower lifetimes due to chronic airtime cost.

V. CONCLUSION

This study developed a join-aware energy model and a pit-conscious control policy for LoRaWAN meters in underground or partially buried chambers. The modelling incorporated OTAA (re)join escalation, Class-A RX1/RX2 listening, and state-current profiles, using energy per successful uplink as the main KPI under reliability, latency, and duty-cycle constraints. A MATLAB-based campaign engine combined log-distance propagation with concrete loss and shadowing, LoRa time-on-air, and an ADR controller that applies an SF floor, tracks link margins, and gates re-joins based on recent coverage. Across chamber-loss scenarios and utility reporting rates, the proposed policy lowered energy per success, reduced unnecessary re-joins, and maintained target PDRs while yielding a right-shifted lifetime distribution compared to fixed-SF and default ADR. These results show that environment-aware ADR guard rails and re-join policing directly address the factors that shorten lifetime in subterranean deployments. The study also offers guidance for utilities: apply margin-tracked ADR with SF floors, account for RX-window costs, and prioritise energy per success over airtime alone. Limitations include the single-gateway abstraction, region-specific channel assumptions, and reliance on simulated penetration statistics. Future work includes field calibration, multi-gateway modelling with explicit interference, a matched NB-IoT comparison, and gateway-placement heuristics for deployment planning.

ACKNOWLEDGMENT

The authors gratefully acknowledge the Rand Water Research Chair for facilitation and support of this research, including discussions on deployment conditions and access to operational constraints that informed the modeling assumptions.

REFERENCES

[1] LoRa Alliance, *LoRaWAN® Specification v1.1*, Oct. 2017.

[2] LoRa Alliance, *LoRaWAN® Regional Parameters*, RP002-1.0.4, Sept. 2023.

[3] Semtech, *LoRa® Modem Designer's Guide*, App Note AN1200.13, Jul. 2013.

[4] Semtech, *LoRa® Modulation Basics*, App Note AN1200.22, 2015.

[5] L. Casals, B. Mir, R. Vidal, and C. Gomez, "Modeling the Energy Performance of LoRaWAN," *Sensors*, 17(10):2364, 2017.

[6] S. Maudet, N. Montavont, and G. Z. Papadopoulos, "Refined Node Energy Consumption Modeling in a LoRaWAN Node," *Sensors*, 21(20):6803, 2021.

[7] T. Voigt, M. Bor, U. Roedig, and J. Alonso, "Mitigating Inter-network Interference in LoRa Networks," in *Proc. EWSN*, 2017, pp. 323–328.

[8] D. Magrin, M. Capuzzo, and A. Zanella, "A Thorough Study of LoRaWAN Performance Under Different Parameter Settings," arXiv:1906.05083, 2019.

[9] M. Al-Gumaei, M. Ali, et al., "A Novel Approach to Improve the ADR Scheme for LoRaWAN," *Electronics*, 11(21):3521, 2022.

[10] M. A. Arshad, J. Pyun, et al., "An Improved Adaptive Data Rate for LoRaWAN Networks," 2020 (preprint).

[11] O. Georgiou and U. Raza, "Low Power Wide Area Network Analysis: Can LoRa Scale?" *IEEE Wireless Communications Letters*, 6(2):162–165, 2017.

[12] R. Kufakunesu, G. P. Hancke, and A. M. Abu-Mahfouz, "A Survey on ADR Optimization in LoRaWAN," *Sensors*, 20(18):5044, 2020.

[13] M. Lauridsen, R. Krigslund, M. Rohr, and G. Madueno, "An Empirical NB-IoT Power Consumption Model for Battery Lifetime Estimation," in *Proc. IEEE VTC Spring*, 2018, pp. 1–5.

[14] A. K. Sultania, T. J. O. A. Le, S. Latré, and J. Famaey, "Energy Modeling and Evaluation of NB-IoT with PSM and eDRX," in *Proc. IEEE WCNC Workshops*, 2018, pp. 1–6.

[15] J. Thrane, K. M. Malarski, H. L. Christiansen, and S. Ruepp, "Experimental Evaluation of Empirical NB-IoT Propagation Modelling in a Deep-Indoor Scenario," arXiv:2006.00880, 2020.

[16] ITU-R, "Recommendation ITU-R P.2109-2: Prediction of Building Entry Loss," Aug. 2023.

[17] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "LoRa Scalability: A Simulation Model Based on Interference Measurements," *Sensors*, 17(6):1193, 2017.

[18] R. Farrell, "Analysis and Enhancement of the LoRaWAN Adaptive Data Rate Scheme," Technical Report, Maynooth Univ., 2022.

[19] A. M. Alghamdi, A. Alwadi, et al., "LoRaWAN Performance Analysis for a Water Monitoring System," *Sensors*, 22(19):7188, 2022.

[20] ITU Radiocommunication Sector, *Recommendation ITU-R P.525-4: Calculation of Free-Space Attenuation*, International Telecommunication Union, Geneva, Switzerland, Feb. 2019.

FLASH - An Alternative Approach To Wideband Spectrum Sensing in 5G Environments

A.B. Jooste[1], A.S.J. Helberg[2]
School for Electrical, Electronic and Computer Engineering
North-West University
Potchefstroom, South Africa
bertusj04@gmail.com[1]
albert.helberg@nwu.ac.za[2]

***Abstract*—Opportunistic spectrum access using cognitive radios requires accurate spectrum-sensing algorithms, of which deep learning based models have shown the best performance, especially at low signal-to-noise ratios [1]. Furthermore, deep learning algorithms are better suited for wideband spectrum sensing than other non-deep learning based algorithms [2]. However, existing deep learning based spectrum-sensing algorithms are too static to be implemented in the dynamic environment associated with 5G cellular networks. This paper proposes deep learning object detection models for spectrum sensing in the dynamic environment of 5G cellular networks. A novel adaptation of the YOLOv4 model that accepts IQ sequence data (FLASH) is trained on a custom 5G dataset that incorporates dynamic allocation of bandwidth parts, channel bandwidth, and subcarrier spacing within FR1 frequencies. The FLASH model has a mean average precision of 94.9% when tested on the custom 5G dataset, demonstrating its potential for broadband dynamic spectrum sensing.**

***Index Terms*—cognitive radio, dynamic spectrum access, object detection, spectrum sensing, SSD, wideband spectrum sensing, YOLO, 5G**

I. Introduction

The Radio Frequency (RF) spectrum, crucial for wireless communication, is a finite resource. The limitations of existing transmitter and receiver technologies, coupled with the fundamental properties of electromagnetic waves, namely the insufficient data capacity of low-frequency waves and the restricted material penetration of high-frequency waves, restrict practical radio frequency utilisation to the range 3 kHz to 3 THz [3].

A. Spectrum Access Policies

The RF spectrum is a shared resource among all wireless transmitting devices, which requires regulation by entities such as Independent Communications Authority of South Africa (ICASA). Organisations such as ICASA, along with counterparts such as the Federal Communications Commission (FCC) in the United States of America and the Office of Communications (Ofcom) in the United Kingdom, implement a Fixed Spectrum Management (FSM) policy, also called a Fixed Spectrum Access (FSA) policy, for the governance of the RF spectrum [4].

The FSM policy divides the RF spectrum into discrete sections assigned to particular applications, in accordance with the International Telecommunication Union (ITU) guidelines. As an example, the frequency range from 88 MHz to 108 MHz is allocated to Frequency Modulated (FM) radio broadcast. FM radio broadcasters are required to have a licence to broadcast on a designated frequency within a defined bandwidth in a geographical area. Regulatory bodies such as ICASA are responsible for the management of transmission permissions and the authorisation of broadcasting entities.

The International Mobile Telecommunications (IMT) bands assigned by ICASA in accordance with the FSM policy are used for cellular networks. A Mobile Network Operator (MNO) may obtain a licence from ICASA granting exclusive usage rights for a designated amount of bandwidth within a specific IMT band. The licence conferred is restricted to the geographical jurisdiction of the issuing regulatory authority. For example, during the 2022 spectrum auction the MNO, Telkom was successfully licensed as the Primary User (PU) of a channel with 20 MHz of bandwidth in the frequency range from 790 MHz to 862 MHz.

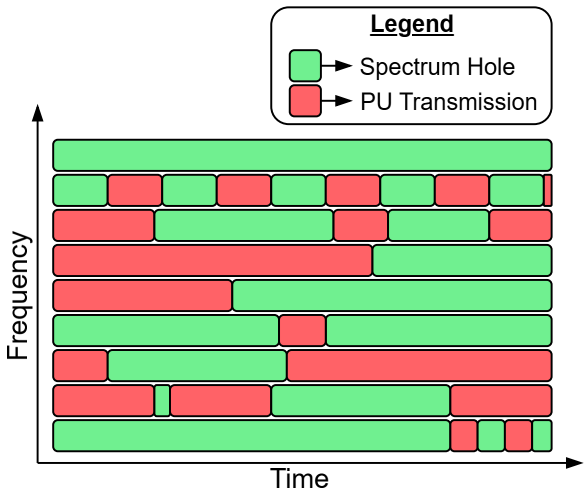


Fig. 1. Illustration of Spectrum Holes

This work is based on the research supported in part by the National Research Foundation of South Africa (Reference Number: FTPM250224299750). Any opinion, finding, conclusion or recommendation expressed in this material is that of the author(s) alone, and the National Research Foundation accepts no liability whatsoever in this regard.

The FSM policy is excellent in preventing interference between RF spectrum users, and the process for a MNO to acquire an exclusive-use licence from a spectrum auction creates competition for access in the highly contested IMT bands. However, the problem with FSM is that it does not optimally use the limited RF spectrum. Since a MNO has exclusive usage rights of a channel in an IMT band throughout the country, if the MNO does not transmit in that band in a certain geographical area, the band would remain idle. Unlike FM radio, another user cannot obtain a licence to transmit on that band in that geographical area, and the MNO channel would simply remain idle. Furthermore, the PU of a channel does not constantly transmit and there are often periods of inactivity. These periods of inactivity where the spectrum remains idle are known as spectrum holes.

In Fig. 1, the vertical axis displays various frequencies, while the horizontal axis depicts time. Each row of rectangles in Fig. 1 signifies a PU's channel. A red shaded rectangle denotes an active transmission by a PU, whereas a green rectangle indicates the channel is currently inactive, constituting a spectrum hole. Should a PU refrain from transmitting in a particular geographical region, their channel remains idle indefinitely, such as the first rectangle in Fig. 1. Spectrum usage can be significantly improved by permitting other users to use the channel when it is idle. Such a user, who accesses a PU's channel without exclusive-use licensing, is referred to as an Secondary User (SU). The practice of enabling an SU to operate on a PU's channel forms part of a Dynamic Spectrum Management (DSM) or Dynamic Spectrum Access (DSA) policy [4] enforced by the national regulator.

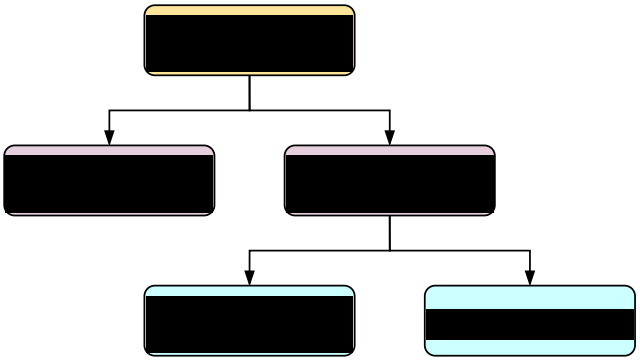


Fig. 2. Dynamic Spectrum Access Hierarchy

Two models that have been proposed to implement a DSM policy are shown in Fig. 2. The first is known as Concurrent Spectrum Access (CSA) and permits a SU to transmit concurrently on a PU's channel whether the channel is idle or not. The CSA model enforces a transmission power restriction on the SU which ensures that the PU only experiences a limited amount of interference from the SU's transmission. The second DSM model is Opportunistic Spectrum Access (OSA), which can be implemented through either a geolocation database or sensing, or a hybrid of both. A geolocation database stores the spectrum usage patterns for a specific geographical area and, therefore, knows which frequencies are idle. An SU can use information in the geolocation database to determine which frequencies it can transmit on. The second method to implement OSA is by using a spectrum-sensing algorithm which constantly senses the RF spectrum and detects spectrum holes. When a spectrum hole is detected by the spectrum-sensing algorithm, the SU can change its transmission strategy to transmit on this spectrum hole [4]. The method of implementing a OSA geolocation database could also require spectrum sensing to determine which frequencies are active in a given geographic region.

B. Cognitive Radio

Cognitive Radio (CR) has been widely recognised as the key technology for enabling DSA. A Cognitive Radio (CR) refers to an intelligent radio system that can dynamically and autonomously adapt its transmission strategies, including carrier frequency, bandwidth, transmit power, antenna beam, or modulation scheme based on the interaction with the surrounding environment and its awareness of its internal states (e.g. hardware and software architectures, spectrum use policy, user needs, etc.) to achieve the best performance. This reconfiguration capability is achieved by the Software Defined Radio (SDR) processor with which the transmission strategies are adjusted by computer software. Moreover, CR is also built with cognition, which allows it to observe the environment through sensing, to analyse and process the observed information through learning, and to decide the best transmission strategy through reasoning [4].

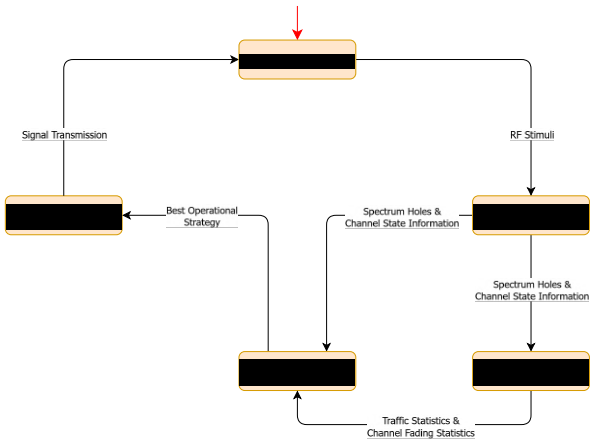


Fig. 3. Cognitive Radio Cycle

A typical cognitive cycle for CR is shown in Fig. 3. An SU with CR capability is required to periodically or consistently observe the environment to obtain information such as spectrum holes in OSA or interference level in CSA. Based on the information collected, it determines the best operational parameters to optimise its own performance subject to the protection of the PUs, and then reconfigures its system accordingly. The information collected over time can also be used to analyse the radio environment, such as traffic

statistics and channel fading statistics, so that the CR device can learn to perform better in future dynamic adaptation.

The spectrum-sensing algorithms that would be implemented in a CR are broadly divided into two categories, i.e., narrowband spectrum-sensing algorithms and wideband spectrum-sensing algorithms. A wideband spectrum-sensing algorithm refers to an algorithm that can detect the presence of more than one PU at a given time rather than being able to only detect a single PU as is the case with narrowband algorithms. Simply put, a wideband spectrum-sensing algorithm would be capable of monitoring nine of the PU channels present in Fig. 1 simultaneously, while a narrowband spectrum-sensing algorithm would only be able to monitor a single channel at a time [2]. Spectrum sensing algorithms are also regarded as either a conventional spectrum-sensing algorithm which does not use deep learning methods or non-conventional spectrum-sensing algorithms which do use deep learning methods. Non-conventional spectrum-sensing algorithms have been shown to often outperform conventional spectrum-sensing algorithms, especially in wideband Long Term Evolution (LTE) scenarios at low Signal-to-Noise Ratio (SNR) levels [2].

Fifth Generation (5G) cellular networks are inherently dynamic, and the amount of bandwidth that a Base Station (BS) uses, how many devices are connected to the BS, the amount of bandwidth used by each User Equipment (UE) and the frequency on which the UEs transmit can constantly change. For example, a single UE can have up to four Bandwidth Parts (BWPs) defined. A UE can switch between its BWPs within a few slots of each frame, which means that the amount of bandwidth used by a UE can vary multiple times in a single frame. The dynamic behaviour of a UE can make spectrum sensing extremely challenging for existing non-conventional spectrum-sensing algorithms since they are static.

C. Static Spectrum Sensing

Examples of non-conventional static spectrum-sensing algorithms are DeepSense [5], DetectNet [6], and Parallel-CNN [7] that all use hypothesis testing to detect spectrum holes. These non-conventional spectrum-sensing algorithms treat spectrum sensing as a multi-label classification problem which is equivalent to hypothesis testing. Therefore, the output of these models is binary, i.e., they either output one if the PU is transmitting on a frequency or zero when the PU is not transmitting. A static spectrum-sensing algorithm senses a fixed number of channels, each defined with a constant bandwidth and a frequency range.

Fig. 4 illustrates how a static spectrum-sensing algorithm works. The red and green rectangles represent the different communication channels in the RF spectrum. The raw In-phase and Quadrature (IQ) samples are fed into the model and the model produces a prediction for each communication channel. If the communication channel is active, the model produces a one, and if the communication channel is inactive, the model produces a zero. An inactive communication channel is equivalent to a spectrum hole, meaning that an SU can transmit on this communication channel.

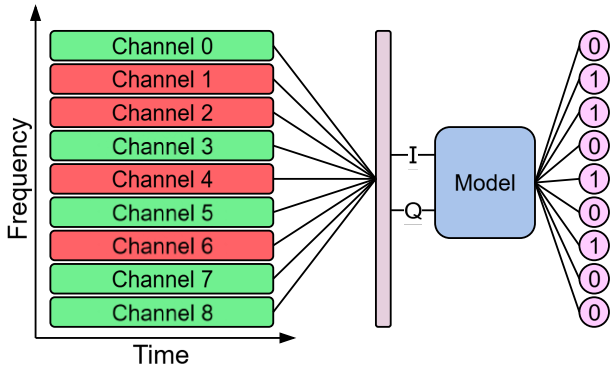


Fig. 4. Static Spectrum-Sensing Model

However, the spectrum in a Fifth Generation (5G) environment would not be as well defined. The number of communication channels in a given Channel Bandwidth (CBW) may vary, the bandwidth of each communication channel may vary, and the start and end point of each communication channel may also vary. Furthermore, even if a model can be trained on all possibilities, the architecture of the model would also need to change based on the current CBW. The output of the model would need to be equal to the maximum number of communication channels available for the given CBW, and in a 5G environment the given CBW can also vary according to demand and region. Therefore, an algorithm that can adapt to these parameters is needed.

D. Dynamic Spectrum Sensing

A dynamic spectrum-sensing algorithm may be able to address the problems mentioned in the previous section. It would work more like an object detector rather than a multi-label classification algorithm. One-stage object detectors, such as the You Only Look Once (YOLO) series of detectors and the Single Shot Detector (SSD) are fast and accurate. They are able to adapt to a varying number of objects in the image (which is equivalent to a varying number of UEs in the CBW) and are also able to adapt to varying image sizes (which is equivalent to a changing CBW). Therefore, with some modification, it may be possible to adapt these object detectors to sense spectrum holes in an RF spectrum rather than objects in an image. This paper investigates the feasibility of using an adapted object detection algorithm as an alternative approach to wideband non-conventional spectrum sensing in a dynamic 5G environment. The paper first implements and verifies popular one-stage object detection models, i.e., You Only Look Once Version 2 (YOLOv2), You Only Look Once Version 3 (YOLOv3), You Only Look Once Version 4 (YOLOv4), and the SSD on a standard object detection dataset, i.e., the PASCAL Visual Object Class (VOC) 2007 dataset. Subsequently, a custom 5G dataset is compiled and used to train the object detectors. The object detectors are trained to identify the active spectrum in the 5G dataset rather than the spectrum holes. If the CBW is known, and the location

of the active spectrum is known, then the location of spectrum holes can be calculated. The best-performing one-stage object detector is identified based on the 5G dataset results. The best-performing model is then slightly adapted, that is, we change its input layer and add a Short-Time Fourier Transform (STFT) layer in order to specialise it for spectrum sensing. The adapted model is nicknamed Frantically Looking At Spectrum Holes (FLASH). If the experimental results show that FLASH is good at spectrum sensing, we will further adapt it in future work. The key contributions of this paper are as follows:

- We introduce an alternative approach to spectrum sensing, in which a modified object detection algorithm is used to detect spectrum holes in a dynamic 5G environment.
- We evaluate the performance of the modified spectrum-sensing algorithm on a custom 5G dataset.

The remainder of the paper is structured as follows: Section II describes the datasets that were used to evaluate the performance of the different models. Additionally, Section II describes how the 5G dataset was generated. Section III details the performance metrics that were used to evaluate the models on. Section IV discusses the object detection methods, i.e. how they were implemented, how they work, as well as the advantages and disadvantages of each. Section IV also describes what would have to be adapted in each model in order for the model to better suit the purpose of spectrum sensing. Section V discusses the results of the experiments; this section also details which model was chosen to adapt and implement as FLASH. Lastly, in Section VI conclusions and recommendations for future work are made.

II. Dataset

Two datasets were used during the experimentation phase of this investigation. The first dataset is a well-known object detection dataset and was used for verification. The second dataset is a custom-generated 5G dataset which simulates the inputs that a spectrum-sensing algorithm would encounter in practice.

A. Verification Dataset

The well-known PASCAL VOC 2007 dataset was used for verification [8]. This dataset has an annotation file for each image in the dataset. The annotation file is an XML file and contains information about its associated image. The most important information in the annotation file for our purpose is the class of all the objects present in the image and the bounding box coordinates for each of the objects in the image. The coordinates of the bounding box in the PASCAL dataset are given in the form (x_1, y_1, x_2, y_2) where (x_1, y_1) represents the upper left corner of the bounding box and (x_2, y_2) represents the lower right corner of the bounding box. However, for the purpose of training an object detector model in MATLAB we need to extract the necessary information from the XML file and convert the bounding box coordinates into the form (x_1, y_1, w, h) where (x_1, y_1) represents the upper left corner of the bounding box and (w, h) represents the width and height of the bounding box, respectively.

We use (1) to calculate the width and height of the bounding box. In the PASCAL VOC dataset, all coordinates are measured in pixels relative to the upper left corner of the image, that is, the upper left corner of the image has the coordinate $(0,\ 0)$.

$$\begin{aligned} w &= x_2 - x_1 \\ h &= y_2 - y_1 \end{aligned} \tag{1}$$

The PASCAL VOC 2007 dataset was selected due to its compact size, which reduced training durations. Moreover, performance metrics for various models evaluated on this dataset are available in their respective publications, providing us with quantitative benchmarks for model performance comparison and verification.

A combined datastore of the "imageDatastore", which contained all the images, and the "boxLabelDatastore", which contained all the bounding box coordinates and class labels, was used as input to training the model. The dataset was randomised and split into three subsets, 80 % of the dataset was used for training, 10 % for validation and the remaining 10 % was used for testing.

B. 5G Dataset

The 5G dataset was generated using MATLAB's 5G toolbox, which ensures compliance with the 5G New Radio (NR) Third Generation Partnership Project (3GPP) standards. This dataset is tailored to emulate the stochastic and variable nature of a 5G environment. Special attention was devoted to maintaining a balanced representation in various transmission combinations to prevent bias in the dataset. By employing uniformly distributed random numbers and a large dataset, any bias was effectively mitigated.

The 5G dataset is restricted to the Frequency Range 1 (FR1) 3GPP frequency bands. Within FR1, there are 15 designated Channel Bandwidths (CBWs), each linked to a specific maximum number of resource blocks corresponding to a particular Subcarrier Spacing (SCS). For each of the CBWs, the maximum number of resource blocks for the corresponding SCS was defined. Subsequently, the factors for each number of resource blocks in an SCS were calculated. Then, one of the factors was randomly chosen for the purpose of determining how many BWPs would be defined for that specific SCS. The size of the Bandwidth Part (BWP), in resource blocks, was determined on the basis of the randomly selected factor. This process was followed for each SCS in each of the 15 defined CBWs. Once a random number of BWPs had been defined for each SCS in a given CBW, randomly selected BWPs would be activated under the condition that they do not interfere with other active BWPs. A set of 2000 random transmissions with a random number of BWPs, SCSs and locations in the CBW were generated for each of the 15 CBWs. Therefore, the dataset consisted of 30 000 spectrogram images.

Fig. 5 is an example of one of the 2000 spectrograms that was generated for a CBW of 5 MHz in the 5G dataset. The legend in Fig. 5 indicates that there are four active UEs, each UE has an SCS of 15 kHz and transmits on two resource

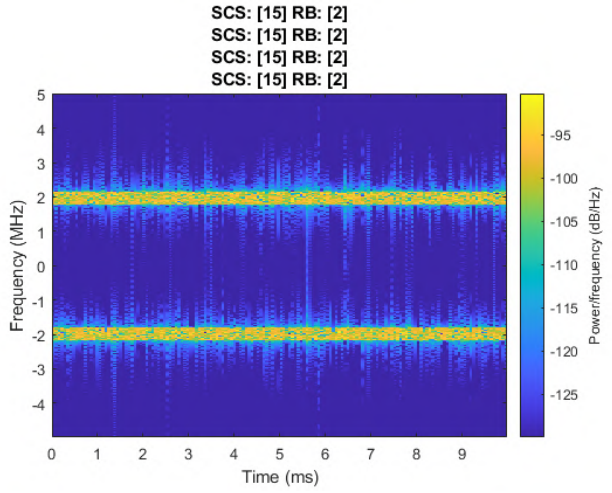


Fig. 5. Example of the 5G Dataset Output for a 5 MHz Channel Bandwidth.

blocks, therefore each UE uses a total of 360 kHz of CBW. Note that the bandwidth shown in the spectrogram is 10 MHz; this is due to the fact that we sample at twice the frequency of the CBW, which is 5 MHz. In Fig. 5 the spectrum is symmetric around 0 Hz. The spectrum is converted to baseband and, therefore, is centred at 0 Hz, but in many cases it is not symmetric around 0 Hz (see e.g. Fig. 10).

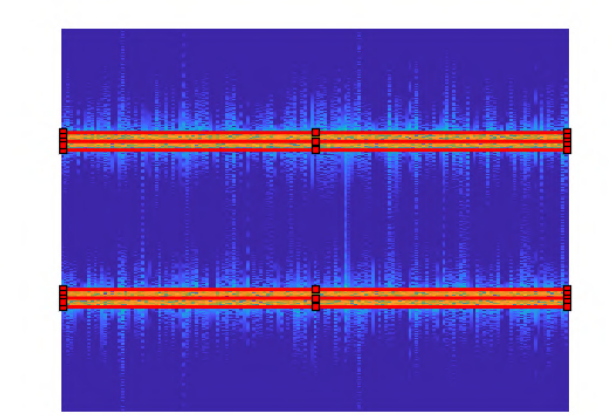


Fig. 6. Example of the 5G Dataset Output for a 5 MHz Channel Bandwidth with Bounding Boxes.

For the purpose of training the different object detectors, we first have to convert the spectrogram to an image. We also have to remove the axis labels, the title, and the colour bar from the spectrogram to create a valid image on which the object detectors can train. Fig. 6 shows an example image that results from this pre-processing. In Fig. 6, we also illustrate the four red bounding boxes, which serve as the ground-truth labels for the detectors.

In order to determine the bounding box of the data within the spectrogram, we first determine the active frequencies by converting the resource blocks from the activated BWPs into their corresponding frequencies values. These values are then mapped to pixels. When we convert the spectrogram into an image, we also define the image size in pixels; therefore, we know that the lowest frequency of the spectrogram would equal pixel row 0 in the image and the maximum frequency would equal the last row of pixels in the image, and since we defined the image size, we can easily map all the other frequencies to pixel values. The transmission spans the entire spectrogram; therefore, the start and end coordinates of the bounding boxes on the x-axis are constants.

We do not vary the SNR of the dataset since we first want to determine if it is possible to implement an object detector for the purpose of spectrum sensing. Additionally, by not varying the SNR levels of the dataset, we reduce the number of variables that affect the performance of the model, and therefore we can better evaluate the validity of the implementation.

The dataset was split into a testing, validation, and training set. The training set consisted of 80 % of the data, the validation set consisted of 10 % and the testing set consisted from the remaining 10 %.

III. Performance Metrics

Performance metrics are tools that are used to evaluate the performance of a model in terms of its accuracy, precision, and recall. This paper focusses on precision and recall, which is used to calculate Mean Average Precision (mAP). mAP is the performance metric that is used to evaluate object detection models.

A. Precision and Recall

Before explaining what precision and recall mean, it is important to first consider what a False Positive (FP), True Positive (TP), False Negative (FN), and True Negative (TN) are and what these terms mean with regard to spectrum sensing. As can be inferred from the name, FP indicates that a value has been incorrectly identified as positive; in terms of spectrum detection, this would mean that the model indicated that the PU is actively transmitting on their frequency, when, in fact, the PU is not transmitting. An FN means that a value has incorrectly been identified as negative, in terms of spectrum sensing this means that the model indicated that the PU is not transmitting on there frequency where in fact the primary user was transmitting. In the realm of spectrum sensing, an FN is more disruptive than an FP since an FN would result in an SU starting to transmit and interfering with the PU's transmission. An FP in spectrum sensing would simply mean that an SU cannot transmit as much as possible and would result in lower efficiency of spectrum usage. The TN and TP values are simply correctly identified values. A confusion matrix is often used to visualise FNs, FPs, TNs, and TPs.

Precision, (Equation 2), is a performance metric that is used to quantitatively measure the frequency with which a model correctly predicts a positive class. Precision answers

the question, "Of all the detections I made, how many were correct?" Precision is often measured on a scale from 0 to 1, where a score of 1 would indicate that the model always predicts the correct target. Precision is used when the cost of an FP is high. However, precision does not take into account FNs and, as stated earlier in spectrum sensing, the cost of an FN is much higher than the cost of an FP.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2}$$

Recall is a performance metric that is used to quantify the frequency with which a model correctly identifies true positives. Recall answers the question, "Of all the actual positive objects that existed, how many did I find?" Recall is used when the cost of a FN is high. Therefore, for the purpose of a spectrum sensing model, more emphasis should be placed on the recall score than on the precision score. Equation (3) shows how the recall is calculated.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3}$$

The relationship between precision and recall embodies an inherent trade-off. Achieving 100 % recall is feasible by classifying everything as an object, thereby encompassing the entire image; however, this results in extremely poor precision. However, high precision can be attained by restricting detections to those with substantial confidence levels, yet this approach leads to a significant number of missed detections, resulting in inadequate recall.

B. Mean Average Precision

Mean Average Precision (mAP) is a key evaluation metric used to quantify the accuracy of a computer vision system. In short mAP quantifies how accurate a computer vision model's prediction is when compared to a ground-truth value. mAP is the standard metric used to evaluate the performance of an object detector. The mAP of an object detector is simply the average of all the average precision values for each class. The average precision of a class is calculated by finding the area under the precision-recall curve. Therefore, a high mAP shows that an object detector has both a high precision and a high recall, which is ideal. The mAP is often defined with a threshold value of 0.50, 0.75, or 0.95, which specifies how much the predicted bounding box needs to overlap with the ground truth box for the predictions to be regarded true. Since it is important for us to reduce the number of false negatives as much as possible, we opt to use a threshold value of 0.50.

IV. Methodology

Object detection models are divided into two broad categories, i.e., one-stage detectors and two-stage detectors. A one-stage detector divides an image into a grid and simultaneously predicts bounding boxes and class scores for each region. A two-stage detector first generates region proposals using a selective search or a region proposal network, and then performs object detection and prediction of the boundary box. One-stage detectors are known as real-time object detectors, as they are much faster at object detection than a two-stage detector.

For the purpose of spectrum sensing we require a real-time object detection model, i.e., a model which is capable of making predictions within a few milliseconds. The most well-known one-stage detectors are the YOLO series detectors and the SSD detector [9] [10] [11] [12] [13]. At the time of writing, 12 versions of YOLO adaptations exist [14]. However, the entire experimental code was written in the MATLAB environment, and MATLAB only implements YOLO versions 1 to 4; therefore, these are the only models that we evaluate. We also evaluate the SSD model, as it set the stage for the newer YOLO models.

Fig. 7 shows a general architecture of an object detector. An object detector receives an image as input; then it sends the image to the backbone, which is responsible for feature extraction. The extracted features are sent to the neck of the object detector, which reshapes the extracted features into the shape required by the head. The head is the output of the object detector and is responsible for providing the bounding box coordinates, the confidence score, and the class predictions in a predetermined format. Not all object detectors have a well-defined neck; however, all object detectors use a backbone and a head.

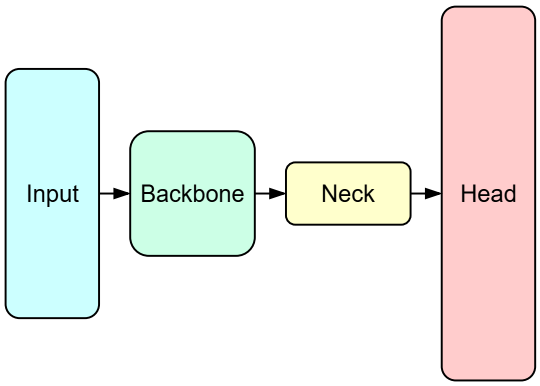


Fig. 7. Object Detection Architecture

YOLOv1 was a fast object detector; however, it was not very accurate. YOLOv1 struggled to detect smaller objects and was not good at generalising. The SSD detector successfully fixed the problems that YOLOv1 had by implementing batch normalisation for improved generalisation and by using pass-through layers, which resulted in higher resolution feature layers at the head of the model which improved detection of smaller objects. Object detectors that use pass-through layers combine predictions from multiple feature maps in the backbone with different resolutions to naturally handle objects of various sizes better. The SSD model also eliminated fully connected layers in its head and implemented a fully convolutional network head [10]. In [10] it was shown that SSD is faster and more accurate than YOLOv1, achieving a mAP of 74.3 % at 59 Frames Per Second (FPS) while

YOLOv1 achieved a mAP of 63.4 % at 45 FPS. YOLOv1 predicted the entire bounding box; however, the SSD detector and subsequent versions of YOLO create a set of default bounding boxes with different aspect ratios and scales per feature map location [10] [11] [12] [13]. The model locates an object by predicting the scores for the presence of an object category in each default box and produces adjustments to the box to better match the shape of the object. The YOLOv2 and YOLOv3 models are very similar to the SSD model, the main difference being that SSD uses the VGG-16 network as backbone whereas YOLOv2 uses DarkNet-19 and YOLOv3 uses DarkNet-53 for their backbones. The loss functions also differ, and how these models implement pass-through layers is also different. YOLOv4 uses the CSP-DarkNet-53 model as its backbone and it also implments a different loss function than the previous YOLO versions.

We first implement these four models in MATLAB on the PASCAL VOC 2007 dataset in order to compare our models' performance with the results obtained in each model's paper. This first step serves as our verification that we implemented the models correctly. Since MATLAB allows us to vary the backbone of each of these models, for YOLOv2 we can either use the "tiny-yolov2-coco" or the "darknet19-coco" backbone, for YOLOv3 we can either use the "tiny-yolov3-coco" or "darknet53-coco" backbone and for YOLOv4 we can choose between the "tiny-yolov4-coco" or "csp-darknet53-coco" backbone. All of these backbones have been trained on the COCO dataset. We decided to implement the "tiny" version of each backbone since this is a smaller network and would be more suitable for implementation on a CR. After the models were validated, we trained the different models on the 5G dataset. The 5G dataset was created such that it is a drop-in replacement for the verification dataset.

The training options that were used to train the models are the same that was used to train the SSD model in [10] as well as the YOLOv1 model in [9]. However, Adaptive Moment Estimation (ADAM) was used as an optimiser instead of Stochastic Gradient Descent With Momentum (SGDM) as ADAM has been shown to converge faster than SGDM. The model was trained for 135 epochs with a batch size of 16. For the first 105 epochs, the learning rate is set at 0.001. The learning rate is then reduced to 0.0001 for the last 30 epochs.

A. FLASH

In order to optimise the object detector for spectrum sensing, we select the best-performing model on the 5G dataset from the four models we tested and alter its architecture. The modified object detector is nicknamed FLASH. For practical implementation in a cognitive radio system, the input should consist of raw IQ samples, and not an image. In MATLAB, we can change the input type from an image to raw IQ samples by changing the "imageInputLayer" into a "sequenceInputLayer" where the sequence consists of the IQ samples. Additionally, the layer following the "sequenceInputLayer" layer expects the data to have a time dimension.

Since an object detection model is designed to work on two-dimensional image data but the raw IQ samples that we send into the model are one-dimensional, we need to reshape the data into two-dimensional data. MATLAB provides a "stftLayer" which calculates the STFT of its input data and outputs two-dimensional data for use in the object detection model.

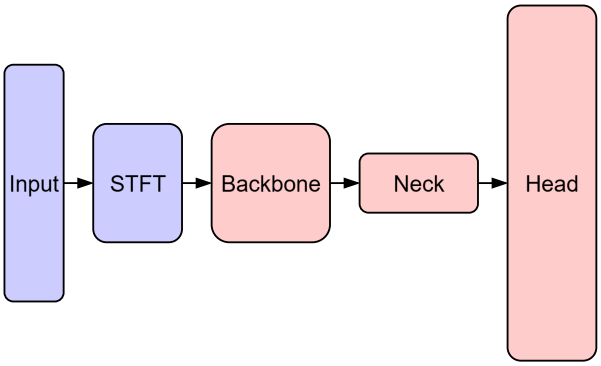


Fig. 8. FLASH Architecture

Fig. 8 shows the architecture of the modified object detector architecture. The purple blocks in Fig. 8 are the modified layers and the red blocks are the same layers of the selected best-performing object detection model.

V. Results

This section describes the results that were obtained from the experiments conducted on the various types of models. The first sub-section describes the verification results, obtained from evaluating the different object detection models on the PASCAL VOC dataset. These results are used to indicate that the models were correctly implemented in MATLAB and that they are able to detect objects in an image. The second subsection shows the results of the unmodified models when they were trained on the custom 5G dataset where the inputs were images of spectrograms and the output are bounding boxes to locate the active spectrum. The third section shows the result of FLASH, which is the modified object detector model. This model takes raw IQ samples as an input rather than images.

A. Verification Results

The results of the verification experiments are shown in Table I. The results of the verification experiments align with the results in each of the models' respective papers [10] [11]. Note that the YOLOv3 [12] and YOLOv4 [13] paper did not evaluate their models on the PASCAL VOC dataset. However, [15] did evaluate the YOLOv3 model on the PASCAL VOC dataset and [16] evaluated YOLOv4. In Table I the "Implemented" column shows the mean average precision of our models whereas the "Target" column shows the mean average precision that the authors published in each models' paper. Our models consistently outperform the results obtained from each model's paper except for the SSD model.

This improved performance can be attributed to the fact that we make use of transfer learning and the backbones we use have been trained on the COCO dataset which is a larger and more complicated dataset than the PASCAL VOC 2007 dataset. However, the SSD detector's VGG-16 backbone has not been pre-trained, therefore its result is closer to the target result. The results shown in Table I are based on a threshold of 0.50 as this is the threshold of the published results.

TABLE I
VERIFICATION RESULTS

Models	Mean Average Precision	
	Implemented	*Target*
YOLOv2	81.6	78.6
YOLOv3	83.4	80.7
YOLOv4	86.5	81.7
SSD	73.5	74.3

The next step was to train these verified models on the 5G dataset in order to determine which model performs the best on that type of data.

B. 5G Dataset Results

The results in Table II show the mAP for each of the models when trained and tested on the 5G dataset. From the table, we can see that the YOLOv4 model outperforms the other modelsmAPFLASH.

TABLE II
MEAN AVERAGE PRECISION FOR MODELS

Models	*mAP*
YOLOv2	82.8
YOLOv3	83.9
YOLOv4	85.1
SSD	68.2

The low score of the SSD model could be attributed to the model struggling to differentiate between two objects that are in close proximity to each other as can be seen in Fig. 5 where two active spectrum components are directly next to each other.

C. FLASH Results

Table III shows the results of FLASH and from the table we can see that changing the architecture of the YOLOv4 algorithm increased the recall and precision of the model.

TABLE III
MEAN AVERAGE PRECISION FOR FLASH

Model	*mAP*
FLASH	94.9

The increase in the recall of the model can be attributed to the use of the STFT layer. This layer contains learnable parameters which the model updates; therefore, it is possible that the model learns what the best configurations to use in order to extract the most information from the raw IQ samples.

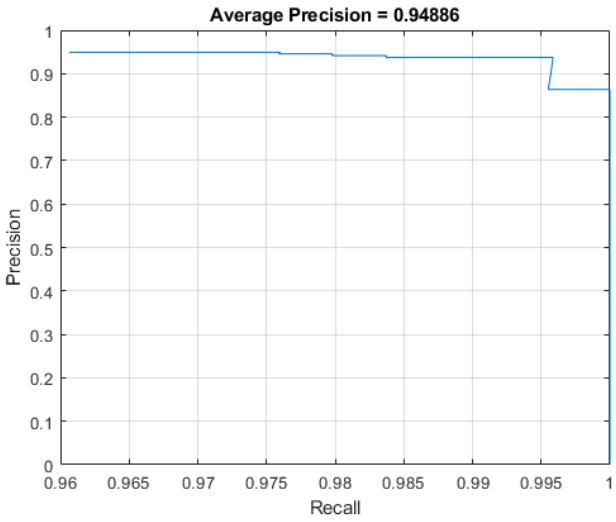


Fig. 9. Precision Recall Curve for FLASH.

Fig. 9 shows the precision recall curve for the FLASH model. Note that since the 5G dataset only contains a single class, the average precision and the mAP are the same. The graph shows that FLASH has high precision and recall, which means that the model is able to correctly detect all PUs in the spectrum.

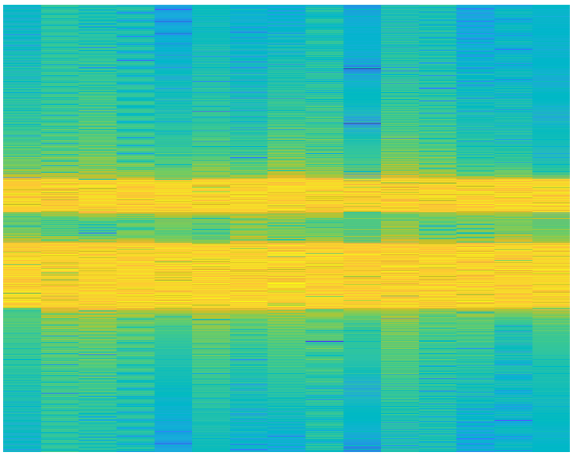


Fig. 10. Primary User Spectrum.

Fig. 10 is a spectrogram formed from raw IQ samples that was fed into FLASH. In this spectrogram we can see that there are currently two active BWPs (yellow bands). One of the BWPs is using a wider bandwidth than the other with a small gap between them. FLASH is trained to detect these active BWPs. Fig. 11 shows where FLASH predicted the active spectra. The red-shade rectangles in Fig. 11 are the labels of each bounding box. In the label of the bounding box is the class of the object which, i.e., PU, as well as how confident FLASH is about the object being a PU. Below the bounding

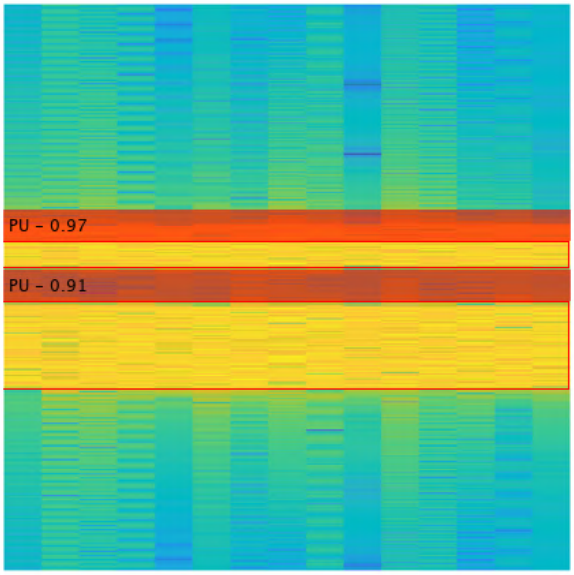


Fig. 11. Predicted Primary User Spectrum.

box labels is a small red rectangle that surrounds the predicted spectrum.

VI. Conclusion

This paper investigated if it is feasible to use an object detection model as a spectrum sensing model, and four different models were evaluated. From the results, it was shown that the YOLOv4 architecture is best suited for this purpose. A novel approach proposed in this paper is the replacement of the image input layers of the model with an IQThe results showed improved detection accuracy of the model in both precision and recall. The model was subjected to various SCS, BWP and CBW yet it was still able to achieve a mAP of 94.9 % showing that an object detector is capable of sensing spectrum in the dynamic 5G environment. Future work would be to adapt the output layer of the model so that it predicts the frequency values of spectrum holes rather than the bounding box coordinates.

References

[1] S. N. Syed, P. I. Lazaridis, F. A. Khan, *et al.*, "Deep neural networks for spectrum sensing: A review," *IEEE Access*, vol. 11, pp. 89 591–89 615, 2023, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3305388. [Online]. Available: https://ieeexplore.ieee.org/document/10217791/ (visited on 05/13/2024).

[2] Y. Arjoune and N. Kaabouch, "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions," *Sensors*, vol. 19, no. 1, p. 126, Jan. 2019, Number: 1 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 1424-8220. DOI: 10.3390/s19010126. [Online]. Available: https://www.mdpi.com/1424-8220/19/1/126 (visited on 04/17/2024).

[3] T. Ratnarajah, C. B. Papadias, and D. T. M. Slock, Eds., *Spectrum sharing: the next frontier in wireless networks*, IEEE series, Hoboken, NJ, USA: Wiley-IEEE Press, 2020, ISBN: 978-1-119-55149-2.

[4] Y.-C. Liang, *Dynamic Spectrum Management: From Cognitive Radio to Blockchain and Artificial Intelligence* (Signals and Communication Technology). Singapore: Springer, 2020, ISBN: 978-981-15-0775-5. DOI: 10.1007/978-981-15-0776-2. [Online]. Available: http://link.springer.com/10.1007/978-981-15-0776-2 (visited on 06/05/2024).

[5] D. Uvaydov, S. D'Oro, F. Restuccia, and T. Melodia, "DeepSense: Fast wideband spectrum sensing through real-time in-the-loop deep learning," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, ISSN: 2641-9874, May 2021, pp. 1–10. DOI: 10.1109/INFOCOM42981.2021.9488764. [Online]. Available: https://ieeexplore-ieee-org.nwulib.idm.oclc.org/document/9488764 (visited on 05/30/2024).

[6] J. Gao, X. Yi, C. Zhong, X. Chen, and Z. Zhang, "Deep learning for spectrum sensing," *IEEE Wireless Communications Letters*, vol. 8, no. 6, pp. 1727–1730, Dec. 2019, Conference Name: IEEE Wireless Communications Letters, ISSN: 2162-2345. DOI: 10.1109/LWC.2019.2939314. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8824091 (visited on 05/31/2024).

[7] R. Mei and Z. Wang, "Deep learning-based wideband spectrum sensing: A low computational complexity approach," *IEEE Communications Letters*, vol. 27, no. 10, pp. 2633–2637, Oct. 2023, Conference Name: IEEE Communications Letters, ISSN: 1558-2558. DOI: 10.1109/LCOMM.2023.3310715. [Online]. Available: https://ieeexplore.ieee.org/document/10236565 (visited on 05/22/2024).

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010, ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-009-0275-4. [Online]. Available: http://link.springer.com/10.1007/s11263-009-0275-4 (visited on 09/20/2025).

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 1063-6919, Jun. 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.

91. [Online]. Available: https://ieeexplore.ieee.org/document/7780460/ (visited on 08/17/2025).

[10] W. Liu, D. Anguelov, D. Erhan, *et al.*, "SSD: Single shot MultiBox detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9905, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2016, pp. 21–37, ISBN: 978-3-319-46447-3 978-3-319-46448-0. DOI: 10.1007/978-3-319-46448-0_2. [Online]. Available: http://link.springer.com/10.1007/978-3-319-46448-0_2 (visited on 09/02/2025).

[11] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, Jul. 2017, Conference Name: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) ISBN: 9781538604571 Place: Honolulu, HI Publisher: IEEE. DOI: 10.1109/CVPR.2017.690. [Online]. Available: http://ieeexplore.ieee.org/document/8100173/ (visited on 09/11/2025).

[12] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *ArXiv*, Apr. 8, 2018. [Online]. Available: https://www.semanticscholar.org/paper/ebc96892b9bcbf007be9a1d7844e4b09fde9d961 (visited on 09/11/2025).

[13] A. Bochkovskiy, C.-Y. Wang, and H. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *ArXiv*, Apr. 23, 2020. [Online]. Available: https://www.semanticscholar.org/paper/2a6f7f0d659c5f7dcd665064b71e7b751592c80e (visited on 09/11/2025).

[14] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, *YOLO evolution: A comprehensive benchmark and architectural review of YOLOv12, YOLO11, and their previous versions*, version: 3, Feb. 25, 2025. DOI: 10.48550/arXiv.2411.00201. arXiv: 2411.00201[cs]. [Online]. Available: http://arxiv.org/abs/2411.00201 (visited on 08/17/2025).

[15] W. A. Ezat, M. M. Dessouky, and N. A. Ismail, "Evaluation of deep learning YOLOv3 algorithm for object detection and classification," *Menoufia Journal of Electronic Engineering Research*, vol. 30, no. 1, pp. 52–57, Jan. 1, 2021, ISSN: 1687-1189. DOI: 10.21608/mjeer.2021.146237. [Online]. Available: https://mjeer.journals.ekb.eg/article_146237.html (visited on 09/19/2025).

[16] M. Zhang, S. Xu, W. Song, Q. He, and Q. Wei, "Lightweight underwater object detection based on YOLO v4 and multi-scale attentional feature fusion," *Remote Sensing*, vol. 13, no. 22, p. 4706, Jan. 2021, Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2072-4292. DOI: 10.3390/rs13224706. [Online]. Available: https://www.mdpi.com/2072-4292/13/22/4706 (visited on 09/19/2025).

A Clustering-Based Multi-Layer Optimization Framework for Energy Efficiency and Congestion in 5G-Enabled IoT Networks

Basil Dlamini
Department of Electrical Engineering
Tshwane University of Technology
Johannesburg, South Africa
dlaminmb@gmail.com

Emmanuel Migabo
Department of Electrical Engineering
Tshwane University of Technology
Pretoria, South Africa
MigaboME@tut.ac.za

Anish M. Kurien
Department of Electrical Engineering
Tshwane University of Technology
Pretoria, South Africa
kurienam@tut.ac.za

***Abstract*— The rapid growth of the Internet of Things (IoT) imposes increasing demands on 5G networks, particularly concerning energy efficiency and interference mitigation. This paper presents a clustering-based, multi-layer optimization framework that models uplink interference among IoT nodes, base stations, and other cellular users. A multi-objective optimization problem is formulated under 5G constraints and solved using an evolutionary algorithm to minimize transmission energy while maintaining communication reliability. Simulation results demonstrate that the proposed method reduces average energy consumption to 120 mJ, compared to 380–660 mJ in existing approaches, and achieves a maximum transmission energy of 90 mJ, versus 220–625 mJ in benchmarks. Additionally, it maintains a 91.5% average packet delivery ratio for nodes at a distance from the gateway, outperforming static clustering (48.0%), mesh (40.3%), FSRC (61.2%), and SFPCR (68.2%). These improvements enhance both energy efficiency and network robustness under realistic deployment conditions.**

Keywords—5G networks, IoT, energy efficiency, interference mitigation, clustering, network coding, congestion control, multi-objective optimization, evolutionary algorithms, quality of service (QoS).

I. INTRODUCTION

The Internet of Things (IoT) is transforming personal, industrial, and societal domains by connecting billions of devices, thereby improving operational efficiency and user experience across sectors such as healthcare, transportation, and manufacturing [1], [5]. In smart cities, for example, real-time, low-latency communication between autonomous systems demands ultra-reliable, high-throughput network infrastructures. However, legacy 4G systems are inadequate for supporting such dense, latency-sensitive deployments due to limitations in congestion control, and spectrum efficiency [3].

To meet these demands, 5G and beyond offer enhanced capabilities through massive machine-type communication (mMTC), ultra-reliable low-latency communication (URLLC), and improved energy and spectral efficiency. Additionally, the Fourth Industrial Revolution (4IR) intensifies the need for scalable, sustainable, and energy-efficient network architectures [2]. However, realizing the full potential of 5G in large-scale IoT deployments requires intelligent and adaptive resource management strategies.

This paper is motivated by the limitations of existing 5G-IoT optimization studies [15], which focus on single-layer or single-objective approaches and consequently face challenges in scalability, energy efficiency, and congestion management in dense IoT environments. To address these gaps, a novel clustering-based, multi-layer optimization framework is proposed to improve energy efficiency, reduce uplink interference, and mitigate congestion. By modelling realistic network behaviour and applying evolutionary multi-objective optimization, it ensures scalable, reliable communication.

A. Problem statement, proposed solution and hypothesis

With the exponential growth of IoT devices in 5G networks, energy consumption and spectrum congestion have emerged as critical challenges. High device density leads to increased interference and inefficient retransmissions, resulting in energy waste and reduced network reliability [6].

Existing optimization methods typically operate at isolated layers (physical, MAC, or network), overlooking cross-layer dependencies inherent in dense 5G-IoT environments. This fragmentation fails to address the compounded issues of congestion and energy inefficiency.

Proposed Solution

This work proposes a multi-layer optimization framework that integrates:

- Dynamic clustering with energy-aware Cluster Head (CH) selection
- Network coding to reduce redundant transmissions
- Mixed-gradient-based multi-hop routing, where CHs select next-hop nodes based on residual energy and link quality

These combined techniques aim to reduce communication overhead, balance load across the network, and extend overall network lifetime.

Research Hypothesis

The proposed multi-layer framework will enhance energy efficiency and reduce congestion in 5G-enabled IoT networks compared to existing single-layer approaches.

Specifically, we hypothesize that:

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

- Average energy consumption will be reduced by ≥ 60%, achieving low values compared to conventional methods.
- Maximum transmission energy per node will be reduced to 90 mJ, from as high as 625 mJ in baseline schemes.
- Packet delivery ratio at extended distances will improve to ≈ 90%, significantly outperforming static clustering (48%), mesh (40%), and SFPCR (68%)

The core research hypothesis is that a multi-layer optimization framework, utilizing dynamic clustering, network coding, and mixed-gradient multi-hop routing, can enhance energy efficiency and reduce congestion in 5G-enabled IoT networks compared to single-layer approaches.

B. Methodology and Procedure

This research adopts a structured methodology comprising five main phases, encompassing modeling, protocol design, algorithm development, and simulation-based performance evaluation. The complete workflow is summarized in Fig. 1.

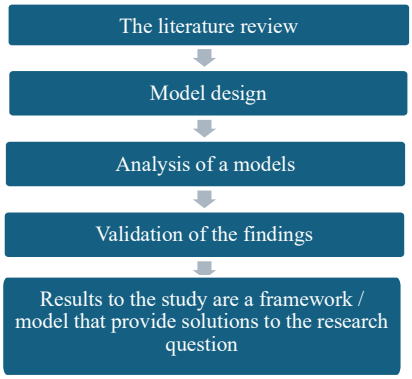


Fig. 1.- Step-by step Procedure towards solving energy and congestion issues

Phase 1: Literature Review and Gap Analysis

The study began with a comprehensive literature assessment and gap analysis of 5G IoT network energy efficiency and congestion control solutions of the current schemes which are :static clustering, mesh topology, SFPCR, and FSRC. This phase identified energy efficiency and congestion mitigation constraints in these systems and promoted a new multi-layer optimization framework.

Phase 2: Optimization Algorithm Design

Genetic Algorithm (GA) class of evolutionary methods was chosen to tackle the multi-objective optimization problem based on literature. GA was chosen for its robustness in complex search spaces, convergence speed, and capacity to handle continuous and discrete variables in 5G-based IoT systems [8].

Candidate network configurations were first generated for optimization. All solutions were evaluated using a fitness function that minimized energy usage and congestion within 5G network restrictions. We iteratively reproduced high-performing solutions using crossover and mutation processes. This evolutionary process persisted until a maximum generation number or convergence threshold was attained.

Phase 3: Simulation and Comparative Analysis

The network was simulated without and with the multi-layer optimization framework at this step. Both cases measured overall energy usage and congestion. The framework's efficacy was assessed by comparing simulation results. The hypothesis was confirmed if the optimized strategy improved both measures.

Phase 4: Benchmarking Against Existing Methods

The suggested optimization framework was compared to Phase 1's four approaches. Performance comparisons assessed energy efficiency and congestion control advantages.

Phase 5: Result Interpretation and Model Validation

The final phase involves a review and interpretation of simulation findings to verify the methodology's resilience. This phase also tested the framework's scalability and stability in various network settings to ensure its suitability for 5G IoT applications.

II. LITERATURE REVIEW

The 5G IoT mobile network is projected to meet essential needs like energy efficiency and pervasive connection for end users [6]. This research focuses on improving the stability and efficiency of the Internet of Things with an emphasis on the 5G cellular network, which will handle a rapid traffic increase. To design an intelligent energy management system that improves IoT communication reliability, research gaps and restrictions related to large-scale, enormous IoT systems are reviewed.

Several Multi-Objective Optimization (MOO) algorithms have been studied to handle IoT network performance trade-offs, including energy consumption.

Despite its computing efficiency, the Weighted Sum Method struggles with non-convex Pareto fronts and is inflexible in dynamic IoT environments [7]. The ε-Constraint and Evolutionary algorithm methods optimize a single objective while using others as constraints [7], but require prior knowledge of constraint thresholds, making them challenging for large-scale, adaptive IoT systems.

A. Current Key Approaches

Enabling reliable IoT communication within 5G cellular networks presents several challenges due to the diverse requirements and characteristics of IoT devices and applications. While 5G offers significant improvements over previous cellular generations, there are still certain obstacles that need to be addressed to ensure efficient and dependable IoT communication [8]. Some of the key challenges include:

• IoT networks are expected to accommodate a massive number of devices, ranging from a few to potentially billions [3] Managing this scale of connectivity requires efficient resource allocation while developing coordination mechanisms to address the challenges. Traditional cellular architecture might struggle to handle so many simultaneous connections without proper optimisation.

• IoT networks can consume a substantial amount of energy hence Balancing the need for reliable communication with energy efficiency is crucial [4][5].

To address these challenges, ongoing research and development efforts are focused on optimising network protocols, designing adaptive resource management strategies, leveraging edge computing for lower latency, developing new modulation and coding schemes, and implementing advanced interference mitigation techniques.

A.1. Existing Energy Optimization Approaches

Energy-efficient communication techniques for 5G with IoT devices have been a significant focus in the research community [3]. These techniques aim to prolong the battery life of IoT devices while maintaining reliable communication. Below are some of the methods that have been explored along with studies demonstrating their impact on energy savings:

• Duty cycling involves periodically turning the radio interface on and off to save energy during idle periods [6]. While effective, it can lead to increased latency. Researchers have worked on optimising duty cycling patterns to balance energy consumption and latency.

• The investigation conducted by academics focuses on exploring communication techniques that are energy-efficient for the IoT devices within the context of 5G networks [6]. The authors offer an adaptive duty cycling technique that considers the traffic pattern of the device and dynamically modifies the duty cycle in response. The findings demonstrate significant reductions in energy consumption while limiting the negative effects on delay.

• Adaptive modulation and coding (AMC) adjust the modulation and coding schemes based on the channel conditions, optimising the data rate according to the signal quality [9]. This technique can significantly reduce energy consumption by adapting to the varying link conditions. Joshua et al. [9] demonstrates that adapting modulation and coding based on channel quality leads to substantial energy savings without compromising data rates.

• Sleep scheduling involves coordinating the sleep and wake cycles of IoT devices to minimise their active periods. By aligning device wakeups with anticipated data transmissions, energy consumption can be reduced. Researchers propose a sleep scheduling mechanism for IoT devices in 5G networks [9]. The study evaluates the trade-off between energy consumption and communication delay, showing that careful sleep scheduling strategies can lead to significant energy savings.

• Cross-layer optimisation involves coordination between the different layers of the communication protocol stack to make energy-efficient decisions [5]. This can include adapting transmission power, modulation, coding, and sleep scheduling based on real-time channel conditions.

The paper explores a multi-layer optimisation approach to enable a 5G communication network that is energy-effective and also demonstrates that considering interactions between different protocol layers can yield substantial energy savings. These studies highlight the potential of energy-efficient communication techniques for IoT devices in 5G networks. However, it is crucial to point out that the effectiveness of these techniques is reliant upon aspects such as the particular IoT application, network circumstances, and the devices involved. As research in this area continues, more refined and application-specific energy-saving methods are likely to emerge, thus contributing to the realisation of efficient and long-lasting IoT deployments within 5G networks [10].

A.2. Multi-layer Optimisation Techniques

Multi-layer optimisation techniques for IoT communication in 5G networks aim to improve overall system performance by considering interactions between different protocol layers. These approaches can lead to enhanced reliability, energy efficiency [1]. Below some studies that propose cross-layer designs and highlight the benefits of such holistic approaches are mentioned.

Some authors optimise both the physical layer parameters and the parameters of the Medium Access Control (MAC) layer simultaneously [16]. The results demonstrate that the proposed cross-layer design improves reliability and energy efficiency by exploiting the synergy between the layers.

An extensive study of cross-layer optimisation strategies for wireless networks, including 5G and beyond, is presented [17]. There is also a strong discussion on the integration of physical, MAC, and network layers to optimise performance metrics such as congestion min imisation and energy efficiency. The study showcases how joint optimisation can lead to improved system performance compared to isolated optimisations in individual layers.

In this research, the authors propose a joint cross-layer design for IoT communications in 5G networks [18]. The study optimises both the physical layer parameters and the Medium Access Control (MAC) layer parameters simultaneously. The results demonstrate that the proposed cross-layer design improves energy efficiency and minimize network congestion by exploiting the synergy between the layers.

The four selected existing schemes

This study primarily uses the following existing approaches to compare performance with the proposed scheme in terms of energy efficiency, network congestion and scalability:

A.2.1 Mesh Network Approach

This scheme is a prevalent method in IoT networks because it enables each node to transmit data to and from other nodes, creating a closely interconnected network. The failure of a single node does not result in the failure of the entire network. The fault-tolerant improved version of this scheme is proposed by Zhao et al. [11]. It comes with the benefit of its resilience to network failure which is its main advantage. Mesh networks enhance scalability through their ability to dynamically adapt to changes and provide effective routing. Conversely, the primary disadvantage of mesh networks is their significant data consumption due to the continuous transmission of messages by nodes. Particularly, in extensive implementations such as 5G networks, this may result in congestion and rapid battery depletion.

A.2.2 Static Cluster Network Approach

The static cluster strategy is a popular method for establishing IoT networks. In this approach, nodes are grouped into clusters and designated CHs manage communication with the base station (BS). The authors of the study, Zhao et al. [11], discuss this method and highlights its ease of implementation as its core benefit. The static cluster technique is well-suited to situations with less dynamic activities or smaller networks due to its minimal computing overhead, which is its primary advantage. The biggest issue with it, is how rigid it is, which often leads to bottlenecks and poor network performance as the network scales up in number of nodes. This is especially true in dynamic 5G environments.

A.2.3 The Fair and Scalable Relay Control Scheme

The performance of the IoT networks is demonstrated to improve by employing the Fair and Scalable Relay Control (FSRC) approach in LoRa-based LPWANs, Lee et al. [12]. This method seeks to enhance scalability and fairness by distributing the load evenly across the network through dynamic relay selection. The FSRC improves the network's dependability by giving each node an equal opportunity to relay data, regardless of its location. By reducing excessive transfers through, the FSRC minimizes congestion more effectively than the static schemes.

A.2.4 The Spreading Factor Partition-Based Clustering and Relaying (SFPCR)

Mugerwa et al [12] investigate multi-hop LoRa networks for the Internet of Things (IoT) and propose the SFPCR method as a way to address the issue of near-far unfairness in an IoT network. By splitting the network into sections based on the spreading factor, this method makes sure that the load is spread out evenly amongst the IoT nodes. The SFPCR method enhances the communication fairness especially for IoT nodes that are farther away from the base station. This enables the networks to scale up a lot better without a very high depreciation in network QoS and QoE performance. In 5G environments with a lot of activities, managing, partitioning and relaying remains a high energy consuming aspect, which influences both total energy use and network performance.

TABLE I. SUMMARY PERFORMANCE COMPARISON

Scheme	Energy Efficiency	Network Scalability	Network Reliability	Complexity
Static Cluster	Low	Low	Moderate	Low
Mesh Network	Low	High	High	High
SFPCR	Moderate	High	Moderate	Moderate
FSRC	High	High	High	High

III. MULTI-OBJECTIVE OPTIMIZATION AND EVOLUTIONARY ALGORITHMS IN IOT NETWORK

The Multi-objective evolutionary algorithms (MOEAs) have gained prominence in recent years for their effectiveness in simultaneously optimizing multiple conflicting objectives, such as enhancing energy efficiency and minimizing congestion. These algorithms are particularly well-suited for complex systems where trade-offs between objectives must be carefully balanced. [13].

The theoretical underpinnings and real-world uses of MOO and EAs in optimizing extensive IoT networks are examined in this section. It emphasizes how these methods provide a strong foundation for balancing trade-offs, guaranteeing quality of service (QoS), and expanding infrastructure to handle the rapidly growing number of connected devices.

A. A Modelling Approach of Energy Minimisation IoT 5G Cellular Network

This paper presents a mathematical framework for the internet of things (IoT) within the context of 5G network standards. It presents a model of the energy consumption characteristics of the IoT, a framework for multi-objective optimisation function seeking to achieve the optimal waiting time for base station (BS) downlink communication and the optimal transmission power for IoT nodes. These enhancements aim to improve the energy efficiency of the network.

The following framework illustrates how the system integrates multiple optimization strategies across layers of the network stack, from physical and MAC layers to network-level decision-making.

A.1. Energy Consumption Modelling

The energy level of the IoT nodes is modelled as per Equation 1.

$$E = \sum_{i=1}^{N} P_i \times t_i \qquad (1)$$

Where:
E is the total energy consumption in Joules,
P_i and t_i are the transmission power and transmission time of node respectively in Watts,
N is the number of nodes.
The congestion level of the IoT nodes is modelled as per Equation 2.
Congestion:

$$C = \sum_{i=1}^{N} \frac{w_R \times R_i}{w_B \times B} \qquad (2)$$

Where:
C is the total congestion,
R_i is the data rate of node in Bits per seconsd,
B is the available bandwidth in Bits per seconds.
Constraints:

$$P_i \leq P_{max}\ ,\ SINR_{\ i} \geq SINR_{\ min} \quad (3)$$

$$L_i \leq L_{max} \quad (4)$$

where:
P_i is the power of of the node i,
P_{max} is the maximum power of the node i,
SINR is the signal to noise ratio of the node i,
$SINR_{min}$ is the minimum signal to nois ratio of the node i,
L_i is the latency or the round of trip delay of the data from the base station to the node i,
L_{max} is the maximum allowable latency.

A.2. The Evolutionary Solution Algorithm

An evolutionary algorithm is used to solve the formulated optimisation problem after the nodes within the IoT network have been initialised. The population is a set of candidate solutions to the optimisation problem. Each solution is represented by a vector of transmission powers, data rates, and transmission times for a given number of IoT nodes. It evaluates the fitness of each solution. The quality of a solution is evaluated using its energy consumption and congestion subject to the constraints. It selects the best solutions. The best solutions are those with the lowest energy consumption and congestion. It creates new solutions. New solutions are created by combining the best solutions in the population. Repeated steps until the stop a criterion is met. The stopping criterion could be a maximum number of generations or a desired level of convergence.
Its implementation requires the following steps:

These steps are summarised in Figure 1.

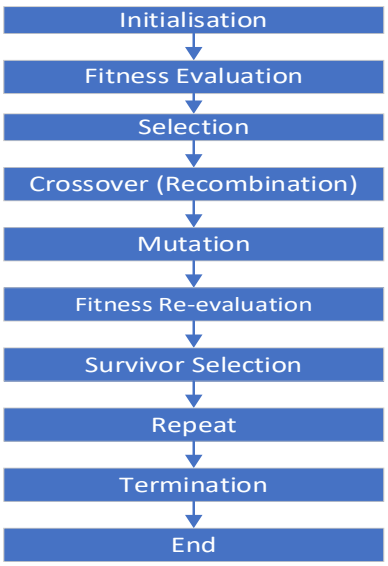


Figure 2 -The Evolutionary Solution Algorithm Steps

A.3. Experimental Setup

The initial number of nodes is used, but there is room for scalability performance evaluation with a lot more IoT nodes. This allows for accurate energy level calculation within available memory and computational resources. The fitness evaluation is based on energy consumption and congestion subject to constraints and the experiment setup population of candidate solutions. The stopping criterion is considered to be a maximum number of generations or desired level of convergence. Table 1 shows the targeted energy consumption performance for a given congestion condition in terms of the expected top five solutions.

B.1. Congestion modeling

This paper proposes an optimization model for IoT systems in 5G networks, targeting reduced BS downlink waiting time and transmission energy. It models congestion and introduces a multiobjective framework.
To reduce congestion, a clustering method is used where Cluster Heads (CHs) are selected based on power availability and node distance. Network coding aggregates data within clusters using linear combinations, lowering packet volume and computational complexity.
The BS decodes these aggregated packets to retrieve original data. Performance is evaluated through average delay and packet loss rate, reflecting network congestion. The approach enhances the scalability and reliability of 5G IoT networks.

B.1.1. A Proposed Clustering and Network Coding Approach to minimise congestion

The objective of using the k-means clustering algorithm in IoT networks is to group similar IoT devices together and then to apply different power, data rate, and transmission time settings to each cluster to minimize the overall energy consumption and congestion in the network [13].

B.1.2. K-means Clustering using 5G Cellular Networks

K-means clustering was employed to group the IoT nodes into clusters based on their location, data rates, or other criteria. Once the nodes had been clustered, different power, data rate, and transmission time settings were applied to each cluster to minimise the overall energy consumption and congestion in the network.
Cluster formation is performed by calculating the three Dimensional (3D) Euclidean distance between each sensor node and the potential CH. Assigning each sensor node to the CH is closest to forming 'K' clusters. The node within each cluster with the highest residual energy and closest proximity to the base station as the CH mist be selected.

B.1.3. Network Coding: Data Aggregation

Each CH collects data from its associated nodes. Each CH aggregates the collected data into a single packet. The energy consumption due to data transmission is modelled as follows:

$$E = d^{\alpha} \times T_{X(Power)} \times Packet_Size \quad (5)$$

where:
E is the energy consumed in Joules,
d is the transmission distance in meters,
α is the path loss exponent, and
$T_{X(Power)}$ is the transmission power in milliwatts.

Packet Size is the size of the data packet in bytes. The amount of energy savings due to data aggregation as compared the direct node to base station data transmissions, is modelled as follows:

$$Energy_{(Savings)} = (N-1) \times E_{direct} \quad (6)$$

where:
$E_{savings}$ is the Energy Savings in Joules,
N is the total number of nodes in the network,
E_{direct} is the energy consumed for direct transmission to the base station.

B.1.3.1. Energy Cost without Clustering and Network Coding

In this approach, each node transmits its data directly to the base station. There is no node playing the role of a cluster head (relay node) for other nodes.

The energy consumption in this approach for each node can be modelled as:

$$E_{node} = {d_{BS}}^{\alpha} \times T_{X(Power)} \times Packet_Size \quad (7)$$

where:

E_{node} is the node residual energy in Joules,
N is the total number of nodes in the network,
$T_{X\ (power)}$ is the transmission power in milliwatts,
${d_{BS}}^{\alpha}$is the transmission distance in meters between the IoT node and the BS,
$Packet_Size$ is the size of the data packet in bytes.

The total energy consumption for the network for N nodes can also be computed as follows:

$$E_{total} = N \times E_{node} \quad (8)$$

where:

E_{total} is the total energy consumed in the IoT network.

- ***Energy Cost with Clustering and without Network Coding***

In this approach, the nodes form each cluster send their packets to their respective cluster heads (CHs). The CHs then forward the packets as they are to the BS, one by one, without aggregating them.

In this approach, the energy consumption of each CH can be modelled as follows:

$$E_{CH} = {d_{BS}}^{\alpha} \times T_{X(Power)} \times \sum_{m=1}^{M} \left(Packet_{Size_m} + Control_{Overhead_m}\right) \quad (9)$$

where:

E_{CH} is the energy consumption of a cluster head node in the IoT network,

$Control_{Overhead_m}$ is the size in bytes of the control overhead information for the packet transmitted by node m.

Considering that there are m other nodes in the cluster other than the CH itself making it a total of m+1 node in the cluster at hand.

On the other hand, the energy consumption for non-CH nodes, can be modelled as follows:

$$E_{nonCH} = {d_{CH}}^{\alpha} \times T_{X(Power)} \times (Packet_{Size}) \quad (10)$$

where:

E_{nonCH} is the amount of energy consumed by non-cluster head nodes within the network.

${d_{CH}}^{\alpha}$ is the transmission distance between the IoT node and the cluster head.

Therefore, the overall total energy consumption is modelled as follows:

$$E_{total} = N_{CH} \times E_{CH} + (N - N_{CH}) \times E_{nonCH} \quad (11)$$

where:

$where$ N_{CH} is the total number of cluster heads in the IoT network.

- ***Energy Cost with Clustering and Network Coding***

In this approach, the CHs apply network coding to aggregate the packets they receive from the other non-CH nodes within the cluster, before forwarding the aggregated packet to the BS

The energy consumption of CHs in this approach is modelled as:

$$E_{nonCH} = {d_{CH}}^{\alpha} \times T_{X(Power)} \times (Coded_Packet_{Size}) \quad (12)$$

Where:

$Coded_Packet_{Size}$ is the size of the coded packet

On the other hand, the energy consumption for non-CH nodes remains the same as per the equation (12).

Therefore, the total energy consumption of the network with N nodes and N_{CH} cluster head nodes, can be modelled as indicated on (12)

Comparison Between Clustering and No Clustering: Clustering typically reduces energy consumption, especially when nodes are far from the base station.
Comparison between Network Coding vs No Network Coding: Network coding can further reduce energy consumption, especially when traffic is high or collisions are frequent, as it can reduce retransmissions.

The outcome of all the above scenarios are compared and the optimum solution is highlighted.

- ***Linear Network Coding***

Each CH encodes data packets from different nodes using linear combinations prior to relaying them to the BS. Upon being received at the base station, the encoded packets are decoded using linear operations to reconstruct the original data.
The encoded packet is modelled as:

$$Y = \sum_{i=1}^{n} d_i G_i \quad (7)$$

where:
$d_i \in GF$
G_i is the individual packet as received from the other nodes in the cluster,
Y is an encoded transmitted packet,
d, is encoding vector or a sequence of coefficients,
n, is the number of packets.

The linear network encoded packets are seen as elements in a finite field. The CH node selects a set of coefficients d_1, d_2, d_3 ... from a defined Galois Field (GF) to linearly encode the received data packets from the other nodes in the cluster. The combined single packet is then transmitted to the BS, therefore, saving energy related to packet transmission. However, this comes with an extra computational overhead which adds to the computational energy cost. Despite this extra computational energy cost, the overall energy consumption is expected to reduce. Amongst the three energy consumption levels within an IoT network, namely the energy consumption due to the hardware, the computational energy cost and the transmission and reception (transceiver) energy costs; the transceiver's one is modelled to be the highest [14]. Therefore, by significantly saving energy due to transmission of packets, despite the increase in computational energy cost, an overall energy consumption reduction is achieved.
This approach is effective only if the base station successfully decodes the encoded packet.
For the BS to be able to decode the encoded packets, it needs to have successfully received at least n encoded packets. The base station decodes the packet by solving a set of n simultaneous equations from the same CH to recover each non encoded packet.
Let us consider Y_j to be a single encoded packet as received by the BS for the n received encoded packets ($1 \leq j \leq n$). Y^j modelled as:

$$Y^j = \sum_{i=1}^{n} d_i^j G_i \quad (8)$$

By solving the set of simultaneous n equations similar to Equation 3.16, where d_i^j and Y^j are known by the BS with n of G_i ($1 \leq j \leq n$) being the unknown, it is possible to recover the decoded packets G_i ($1 \leq j \leq n$).

where:
$d_i \in GF$
G_i is the individual packet as received from the other nodes in the cluster,
GF is a Galois field or a finite field of order 2^n,
Y is an encoded transmitted packet,
Y^j, is a packet. Received,
d, is encoding vector or a sequence of coefficients,
n, is the number of packets.

C. The Proposed Congestion and Energy Mitigation Approach

Linear network coding reduces the quantity of transmissions, alleviating network congestion while minimising enegy usage [2]. Decoding can be performed at multiple nodes, further reducing congestion. The mathematical equation for the energy savings and congestion mitigation are provided as follows:
The total energy savings can be calculated by finding the sum of the energy savings for each cluster:

$$Total\ Energy\ Savings = \sum_{p=1}^{P} (Energy_{Savings})_p \quad (9)$$

where:
$Total\ Energy\ Savings$ is the total energy saved in Joules,
$(Energy_{Savings})_p$ is the energy saved in the p^{th} cluster and
P the total number of clusters in the network.
The reduction in network congestion can be quantified by measuring the average delay and packet loss rate. A decrease in average delay and packet loss rate indicates a reduction in congestion.
In the proposed system, the clustering IoT devices were based on their location to reduce the transmission power required to reach all the devices in the network. The IoT devices were based on their data rates to balance the load on the network and reduce congestion. Network coding was used to reduce the amount of data that needs to be transmitted, which is especially effective for IoT applications that transmit a large amount of redundant data such as sensor data. By combining these techniques, we significantly reduced the energy consumption and congestion in the IoT networks.
The experiment was first set up to cluster IoT devices based on their location. Then, we used network coding to reduce the amount of data that need to be transmitted between the devices in each cluster. Finally, we applied a multilayer optimization technique to further optimize the power, data rate, and transmission time settings for each cluster.
This approach is used to improve the performance and efficiency of IoT networks, enabling them to support more devices and applications.

C.1 Model Design

The next step is to design a model of the 5G network in the discrete network simulator. The model includes the following components:

- IoT devices grouped in network cluster, each with a device selected as CH.
- Base stations (BSs).
- The network channel.

The model is also configured to reflect the given parameters such as the sub-6 GHz mid bandwidth, data rates 10 to 1,000 Kbps, SNR of at least 20 dB, maximum delay 100 ms, and maximum transmission power of P_{max}.

C.2 Implementation of the IoT Clustering and Network Coding Procedures

Once the model has been designed, the next step is to implement the cluster and coding procedure in the discrete

simulator. The IoT clustering and network coding techniques should be implemented in a way that is consistent with the literature review.

C.3. Performance Assessment

Once the IoT clustering and network coding techniques have been implemented, the next step is to conduct a performance assessment. The performance assessment should measure the following metrics:

• Network energy usage.
• Network congestion.

The performance assessment should be conducted with and without the IoT clustering and network coding techniques to provide a comparative analysis of the performance of the network under both conditions.

The process is summarized in Figure 2.

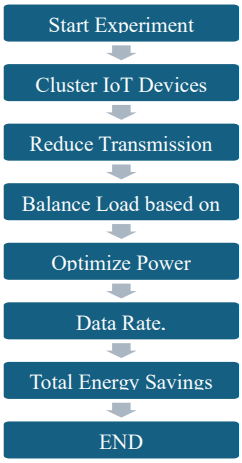


Figure 3 -A Summarized process of congestion mitigation

C.4 Implementation of the Multilayer Optimization Technique

Once the model has been designed, the next step is to implement the multilayer optimization technique in the discrete network simulator. The following steps are taken:

- A set of MATLAB simulation scripts are written to interact with the network emulator through APIs or command-line interfaces.
- The obtained optimized decision variables are passed from the optimizer to the network emulator script to generate the optimized network configurations.
- The energy, congestion and latency performance metrics are then extracted from the simulator output and fed back to the optimizer as fitness values.

IV. OBTAINED RESULTS AND DISCUSSION

After simulating various performance analysis scenarios, the proposed multilayer optimization approach and the static cluster scheme, mesh scheme, SFPCR optimization approach, and FSRC are presented, compared, and analysed. Performance comparison and analysis metrics include network trustworthiness computed by energy consumption variation. The results are then compared to the stated problems to see if the research objectives are met.

The following simulation parameters are used to compare the proposed technique to the selected existing approaches based on the 3GPP 5G network standard.

TABLE II. SIMULATION PARAMETERS

Parameter	Values/Range
Maximum Transmission Power (mW)	31 dBm (Power Class PC1)
Maximum Allowed Traffic Throughput (Gbps)	20 Gbps
Maximum allowed SINR	20 dB
Network topology	Cellular
Mobility of nodes	No mobility (Static)
Maximum Acceptable Latency (seconds)	1 ms
Distance from Gateway (Km)	Ranged from 1 km to 10 km
Number of Devices	Ranged from 10 to 1000
Traffic Payload Size	Ranged from 100 Mbits to 1 Gbits
Simulation duration	1 hour to 24 hours
5G Frequency band	Sub-6 GHz (FR1): 450 MHz to 6 GHz IoT band used: 600 MHz to 900 MHz
Link Layer Protocol	FDMA Sub-band: 200 kHz
Transport Layer	TCP (security, reliability)
Application Layer	MQTT
PHY layer	OFDM
Channel model	Carriers: 868 MHz, 434 MHz AWGN (Additive White Gaussian Noise).

The results obtained are as follows:

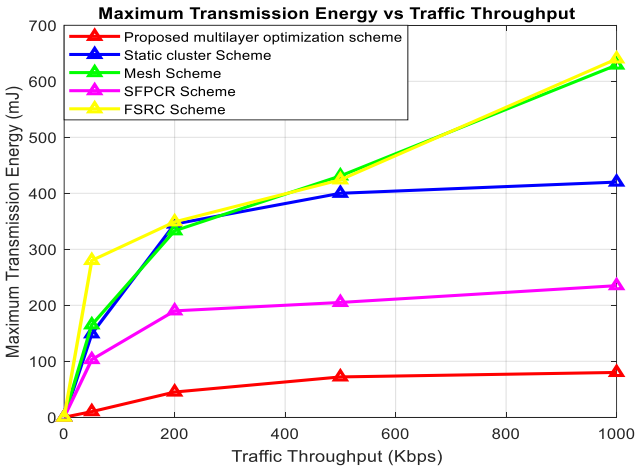


Figure 3 -Maximum Transmission Energy vs Traffic Throughput

Existing 5G-IoT schemes like SFPCR, Static, Mesh and FSRC suffer from high energy use and limited scalability due to their single-layer design. To overcome these issues, this paper proposes a multi-layer clustering-based optimization framework that improves energy efficiency, reduces uplink interference, and mitigates congestion in dense IoT networks.

Figure 3 shows that transmission energy increases with throughput across all schemes in a 10 to 100 node 5G IoT network. The proposed multilayer optimisation scheme achieves a maximum transmission energy of 90 mJ, which is lower than the static cluster (220 mJ), mesh (410 mJ), SFPCR (610 mJ), and FSRC (625 mJ) schemes. This result highlights the scheme's improved energy efficiency under varying traffic conditions, reflecting better resource utilisation and enhanced network lifetime, making it a strong candidate for energy-efficient 5G IoT deployments.

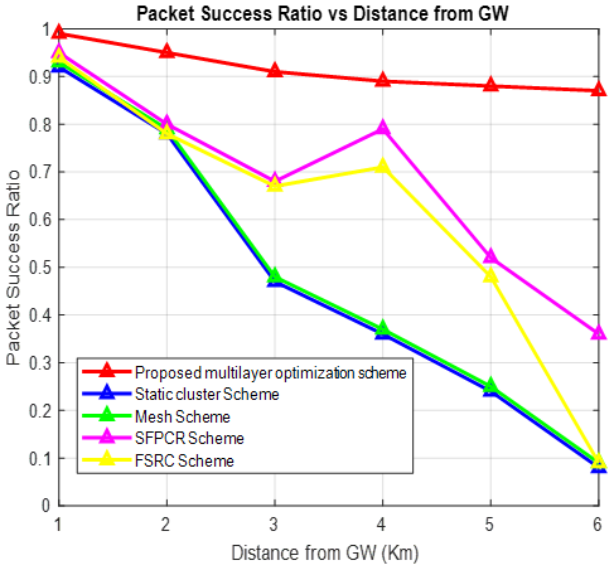


Figure 4 -Packet Success Ratio vs Distance from GW

Figure 4 illustrates the average packet success ratio across varying distances between the gateway and IoT nodes. The proposed multi-layer optimization scheme achieves a high delivery ratio of 91.50%, outperforming static cluster (48.00%), mesh (40.33%), FSRC (61.17%), and SFPCR (68.17%) schemes, demonstrating improved reliability over longer distances. On average, it achieves approximately 37% higher packet success than existing methods. Key advantages include reduced congestion via optimized two-hop communication, cost efficiency by minimizing additional infrastructure, and consistent performance in dense environments, making it ideal for high-demand IoT applications requiring reliable, scalable communication.

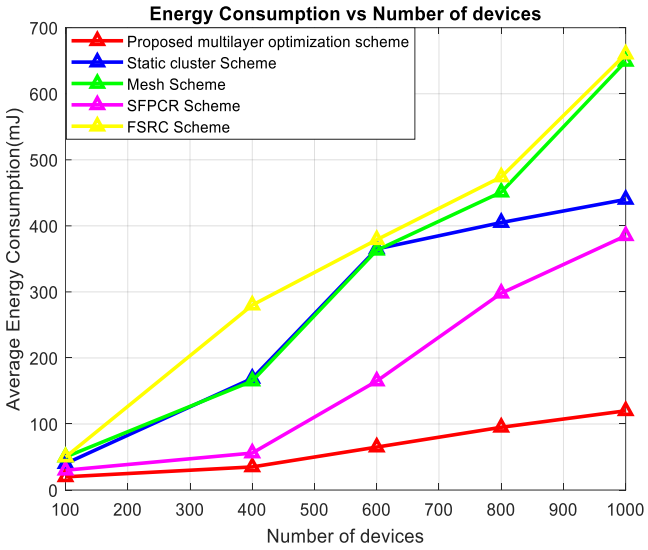


Figure 5 – Energy consumption vs Number of devices

Figure 5 demonstrates that the proposed multilayer optimisation scheme consistently achieves lower average energy consumption as the network scales, recording 120 mJ compared to 380 mJ to 660 mJ for the existing schemes. This highlights its superior scalability and energy efficiency under increasing device counts. The performance gains are attributed to effective clustering and network coding, which enable efficient packet aggregation and node grouping. While the approach depends on successful decoding and optimal cluster head selection, even with occasional retransmissions, energy consumption remains lower if retransmissions stay within the average cluster size. Overall, the multilayer optimisation scheme proves to be a strong candidate for energy-efficient IoT deployments, particularly in large-scale networks with long transmission distances and rising device density.

V. CONCLUSION

The proposed multilayer optimisation technique effectively reduces both network congestion and energy consumption, resulting in an extended network lifetime and an overall improvement in user experience. This ensures that network performance remains within acceptable quality thresholds, even as the number of connected IoT devices continues to grow within 5G environments. The study has successfully addressed the defined research questions and met its objectives by integrating hybrid clustering with network coding to minimise transmission distances and optimise energy usage at cluster heads. Additionally, by aggregating data and optimising spectrum utilisation, the approach significantly alleviates congestion, reduces latency, and limits retransmissions. Overall, the proposed technique enhances energy efficiency, supports scalable and reliable communication, and lays a solid foundation for future research and development in next-generation IoT networks, including the transition toward 6G.

ACKNOWLEDGMENT

The authors would like to thank the Telkom Centre of Excellence (CoE) for their continuous support.

REFERENCES

[1] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: a review," Journal of Big Data, vol. 6, no. 111, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0268-2

[2] B. Alhayani et al., "5G standards for the industry 4.0 enabled communication systems using artificial intelligence: perspective of smart healthcare system of Things Industry 4.0 Interference management Resource optimization," Applied Nanoscience, vol. 3, 2022. doi: 10.1007/s13204-021-02152-4.

[3] X. Liu and X. Zhang, "Rate and Energy Efficiency Improvements for 5G-Based IoT with Simultaneous Transfer," IEEE Internet Things J., vol. 6, no. 4, pp. 5971–5980, 2019, doi: 10.1109/JIOT.2018.2863267.

[4] M. Baniata, H. T. Reda, N. Chilamkurti, and A. Abuadbba, "Energy- efficient hybrid routing protocol for IoT communication systems in 5G and beyond," Sensors (Switzerland), vol. 21, no. 2, pp. 1–21, 2021, doi: 10.3390/s21020537.

[5] D. Zhang, Z. Zhou, S. Mumtaz, J. Rodriguez, and T. Sato, "One Integrated Energy Efficiency Proposal for 5G IoT Communications," IEEE Internet Things J., vol. 3, no. 6, pp. 1346–1354, 2016, doi: 10.1109/JIOT.2016.2599852.

[6] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges," IEEE Access, vol. 6, pp. 3619–3647, 2017, doi: 10.1109/ACCESS.2017.2779844.

[7] M. A. Farag, M. A. El-Shorbagy, A. A. Mousa, and I. M. El-Desoky, "A new hybrid metaheuristic algorithm for multiobjective optimization problems," International Journal of Computational Intelligence Systems, vol. 13, no. 1, pp. 920–940, 2020.

[8] B. Pradhan, V. Vijayakumar, S. Pratihar, D. Kumar, K. H. K. Reddy, and D. S. Roy, "A genetic algorithm based energy efficient group paging approach for IoT over 5G," Journal of Systems Architecture, vol. 113, p. 101878, 2021.

[9] J. O. Ogbebor, A. L. Imoize, and A. A. A. Atayero, "Energy Efficient Design Techniques in Next-Generation Wireless Communication Networks: Emerging Trends and Future Directions," Wirel. Commun. Mob. Comput., vol. 2020, 2020, doi: 10.1155/2020/7235362.

[10] I. A. Najm, A. K. Hamoud, J. Lloret, and I. Bosch, "Machine learning prediction approach to enhance congestion control in 5G IoT environment," Electron., vol. 8, no. 6, 2019, doi: 10.3390/electronics8060607.

[11] Y. Zhao, K. Liu, X. Xu, H. Yang, and L. Huang, "Distributed Dynamic Cluster-Head Selection and Clustering for Massive IoT Access in 5G Networks," Applied Sciences, vol. 9, no. 1, p. 132, 2019. [Online]. Available: https://doi.org/10.3390/app9010132.

[12] D. Mugerwa, Y. Nam, H. Choi, Y. Shin, and E. Lee, "SF-Partition-Based Clustering and Relaying Scheme for Resolving Near–Far Unfairness in IoT Multihop LoRa Networks," Sensors, vol. 22, no. 23, p. 9332, 2022. [Online]. Available: https://doi.org/10.3390/s22239332.

[13] M. Polese, M. Giordani, T. Zugno, A. Roy, S. Goyal, D. Castor, et al., "Integrated access and backhaul in 5G mmWave networks: Potentials and challenges," IEEE Commun. Mag., vol. 58, no. 3, pp. 62-68, Mar. 2020.

[14] C. Han, T. Harrold, S. Armour, I. Krikidis, S. Videv, P. M. Grant, H. Haas, J. S. Thompson, I. Ku, and C. X. Wang, "Green radio: Radio techniques to enable energy-efficient wireless networks," IEEE Commun. Mag., vol. 49, pp. 46–54, 2011, [CrossRef].

[15] L. Chen et al., "Energy-aware clustering for massive IoT in 5G using evolutionary algorithms," *IEEE Internet of Things Journal*, vol. 11, no. 4, pp. 3340–3352, 2024.

[16] A. M. Ramly, N. F. Abdullah, and R. Nordin, "Cross-layer design and performance analysis for ultra-reliable factory of the future based on 5G mobile networks," IEEE Access, vol. 9, pp. 68161–68175, 2021, doi: 10.1109/ACCESS.2021.3078165.

[17] A. Sahbafard, A. Springer, P. Popovski, and H. P. Bernhard, "Scalable Uplink Modeling for Resource Management in 5G URLLC Networks," in *Proc. 2024 IEEE 20th International Conference on Factory Communication Systems (WFCS)*, Toulouse, France, Apr. 2024, pp. 1–5, doi: 10.1109/WFCS60972.2024.10540945.

[18] S. M. A. Rizvi, U. Khalid, S. Chatzinotas, T. Q. Duong, and H. Shin, "Controlled Quantum Semantic Communication for Industrial CPS Networks," *IEEE Trans. Netw. Sci. Eng.*, vol. XX, no. XX, pp. XXX–XXX, 2025, doi: [DOI here].

Validation of an OpenAirInterface 5G Testbed for Uplink-Heavy Live Audio-Visual Production

1st Andries Oelofse, 2nd Melvin Ferreira, 3rd Herman Blackie
School of Electrical, Electronic and Computer Engineering
North-West University
Potchefstroom, South Africa
33683654@mynwu.ac.za[1], melvin.ferreira@nwu.ac.za[2], herman.blackie@nwu.ac.za[3]

***Abstract*—Live Audio-Visual Production (LAVP) requires wireless camera connectivity to provide operators with mobility and flexibility when capturing dynamic events. Fifth-generation (5G) networks, particularly when deployed with customized Time Division Duplexing (TDD) configurations, present a promising platform for such uplink-heavy use cases. This paper investigates the suitability of the open-source OpenAirInterface (OAI) platform as the foundation for a 5G testbed targeting LAVP. We established a controlled and repeatable test environment using a cabled RF setup to evaluate throughput across multiple TDD slot configurations. Measured results were compared against theoretical 5G throughput calculations and findings from related work. Using OAI RAN v2.3.0, downlink performance closely matched theoretical maximums, whereas uplink throughput was significantly underutilized. Spectrum analysis revealed a scheduling bug preventing the use of all uplink slots in configurations with extended TDD periodicities. After updating to the latest development branch (2025.w37), uplink throughput improved by 60.95%, confirming the bug fix. However, performance remains below theoretical limits, highlighting persistent challenges in uplink-heavy deployments. This work contributes by validating a reproducible OAI-based testbed, identifying and confirming a critical uplink scheduling issue, and underscoring the need for rigorous version-specific validation in actively developed open-source 5G platforms.**

***Index Terms*—5G, OpenAirInterface, Live Audio-Visual Production, Uplink, TDD, Testbeds**

I. Introduction

Live media is increasingly popular due to the greater social interactivity it provides compared to pre-recorded content [1]. Its real-time nature enables audiences to share experiences simultaneously and engage with events as they unfold. The production of such content, referred to as live audio-visual production (LAVP), demands flexibility since no retakes are possible once content is broadcast. Wireless camera technologies play a critical role in LAVP by allowing operators freedom of movement and the ability to capture dynamic, on-the-spot footage.

Current wireless camera solutions for LAVP rely on coded orthogonal frequency-division multiplexing (COFDM), Wi-Fi, and 5G, each offering distinct trade-offs in terms of reliability, throughput, and cost. Prior work, such as [2], has demonstrated the use of 5G in multi-camera studio environments, but largely within proprietary 5G core and radio access network (RAN) ecosystems provided by commercial vendors.

In contrast, we argue that open 5G platforms, such as the OpenAirInterface (OAI) project, offer unique advantages in accessibility, transparency, and cost-effectiveness for video transport in LAVP. This paper investigates and validates the current capabilities of OAI as a 5G platform for uplink-heavy scenarios requiring customized Time Division Duplex (TDD) configurations. Our work builds on an implemented LAVP-specific testbed to quantify performance, benchmark against theoretical throughput limits, and compare with similar testbeds from related research.

The main contributions of this paper are threefold:

1) We provide complete configuration files and raw measurement data to support reproducibility and cross-validation of other testbeds.
2) We identify uplink throughput issues in the OAI platform and report them to the OAI community.
3) We validate our testbed against theoretical maximums and related work, highlighting both strengths and persistent limitations in uplink-heavy use cases.

The remainder of the paper is organized as follows: Section II reviews related 5G standalone (SA) testbeds based on the OAI software stack. Section III introduces TDD slot configuration and 3rd Generation Partnership Project (3GPP) throughput calculations. Section IV describes our testbed setup, followed by methodology in Section V. Results from OAI RAN v2.3.0 are reported in Section VI, while updated results using integration 2025.w37 are presented in Section VII. Finally, Section VIII summarizes our findings and discusses current challenges in uplink throughput on OAI.

II. Related Work

Our review focuses on 5G SA testbeds built on the OAI software stack, as this aligns with the platform used in our study.

This work is supported by the Sentech Research and Innovation Collaborative Network and Telkom Centre of Excellence at the North-West University.

The work in [3] was initially of interest due to its focus on real-time video streaming. Its most valuable contribution, however, was the description of a testbed using coaxial cables and radio frequency (RF) power splitters to connect to a Quectel RM500Q-GL modem. This approach enabled repeatable measurements unaffected by environmental variability. The authors also conducted an over the air (OTA) campaign testing multiple TDD configurations with a smartphone as the user equipment (UE), but only downlink performance was considered.

Similarly, [4] employed coaxial connections as the radio interface, using two Universal Software Radio Peripheral (USRP) N310s, one configured as a gNodeB (gNB) and the other as a UE. The TDD configuration was not explicitly stated but appears to be OAI's default downlink-heavy pattern of seven downlink and two uplink slots.

Other studies opted for OTA measurements with a 30 kHz sub-carrier spacing (SCS). Most are assumed to have used OAI's default TDD configuration, as no alternatives were specified. Each had a different emphasis but collectively informed our understanding of OAI performance and configuration.

In [5], a USRP B210 served as the gNB with various smartphones as UEs over a 20 MHz channel. The focus was on comparing different core and RAN technologies. However, we note that the throughput comparison was not entirely fair, since srsRAN employs a different default TDD pattern (six downlink and three uplink slots).

Reception characteristics of OAI using a USRP X310 as the gNB and a smartphone as the UE were examined in [6]. Meanwhile, [7] and [8] contributed detailed insights into frequency and network configurations of the OAI RAN and core, respectively, both using a Quectel RM500Q-GL modem as the UE.

A broader investigation into throughput, latency, and reception with OAI was presented in [9], using a USRP B210 and an N310 as radio units (RUs). Both tested configurations used a symmetrical TDD pattern with a 2.5 ms periodicity, offering useful insight into OAI's potential for low-latency, uplink-heavy scenarios. This aligns more closely with the requirements of LAVP compared to other studies.

Finally, [10] investigated reproducibility across testbeds with identical software defined radios (SDRs), modems, and RAN configurations but different computer hardware and software. Common hardware included the USRP B210 and Quectel RM520Q-GL. Multiple TDD and bandwidth configurations were tested to study throughput and one-way delay across sites. Results showed that reproducibility was not guaranteed, possibly due to unspecified physical environments (e.g., OTA vs. Faraday cage vs. coaxial). Because this work reports diverse slot configurations with corresponding throughput results, it provides a valuable baseline for comparison with our measurements.

III. Background

A. TDD pattern and periodicity

TDD pattern configuration is one of the methods that allows 5G to be such a flexible standard. It allows a network to be biased towards more uplink or downlink heavy configurations as well as high-throughput or low-latency configurations, depending on what the use case requires. A 5G frame always has a 10 ms duration and is divided into slots that can be assigned as an uplink, downlink, or a special slot that acts as a transition between downlink and uplink. This special slot leaves a gap of unallocated guard symbols in the middle of the slot to account for the propagation delay of the signal from the UE to the gNB.

Fig. 1 shows two sections of 5G frames with different TDD configurations. The first Fig. 1(a) has a configuration where uplink and downlink are evenly prioritised. This is done by symmetrically configuring two uplink and two downlink slots within the 5-slot TDD periodicity, leaving one space for the special slot. The 5-slot periodicity means that the defined pattern is short but is repeated multiple times within a single frame. Consequently, such a configuration would have lower latency due to the more regular transmissions occurring between the UE and gNB.

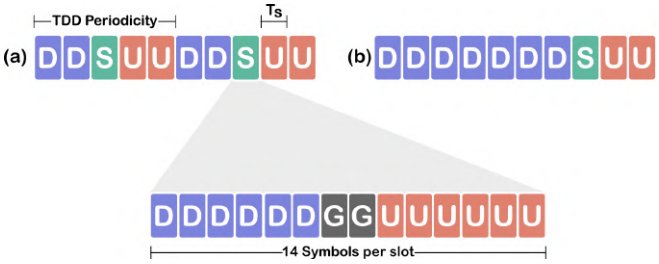


Fig. 1. Example of two different TDD configurations, (a) balanced symmetrically, and (b) biased toward downlink.

Within the second configuration Fig. 1(b), more slots are allocated as downlink, and a longer periodicity of 10 slots is used. This prioritises downlink throughput by adding some latency, but wasting less time on special transition slots, which could be used to transmit more downlink information.

Note that the format of the symbols in the special slot can also be modified to bias toward uplink or downlink. However, a minimum guard period of two symbols should always be maintained.

B. Theoretical Maximum Throughput

To calculate the maximum throughput for a given 5G RAN configuration, the 3GPP TS 38.306 specification [11] defines (1). For our application, $J = 1$ (single component carrier), $f^{(j)} = 1$ (scaling factor), and $R_{\text{max}} = 946/1024$ (highest code rate for error correction). Furthermore, $v_{\text{Layers}}^{(j)} = 1$ (maximum supported multiple-in multiple-out (MIMO) layer), $Q_m^{(j)} = 8$ (maximum modulation order), and $\text{OH}^{(j)}$ is the overhead factor: $\text{OH}^{(j)} = 0.14$ for downlink and $\text{OH}^{(j)} = 0.08$ for uplink. This formula calculates throughput

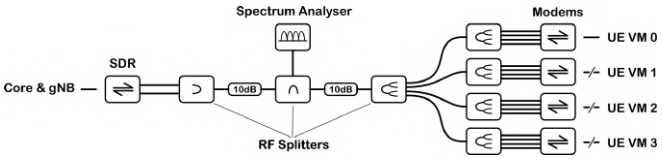


Fig. 2. Interconnection of all RF components in the testbed.

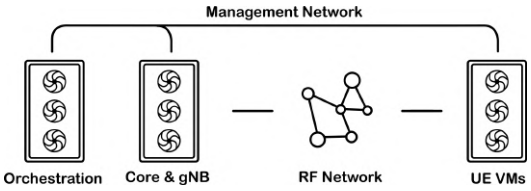


Fig. 3. Role distribution across machines in the testbed.

assuming the full frame is allocated to either downlink or uplink; therefore, it must be scaled by the symbol ratio for downlink or uplink.

$$\text{Data rate (Mbps)} = 10^{-6} \cdot \sum_{j=1}^{J} \left(v_{\text{Layers}}^{(j)} \cdot Q_m^{(j)} \cdot f^{(j)} \cdot R_{\max} \cdot \frac{N_{\text{PRB}}^{\text{BW}(j),\mu} \cdot 12}{T_s^{\mu}} \cdot (1 - \text{OH}^{(j)}) \right) \tag{1}$$

By substituting the constant variables and including the uplink-to-downlink ratio, the downlink and uplink throughput equations are derived as in (2) and (3), respectively. Here, $N_{\text{PRB}}^{\text{BW},\mu}$ is the number of Physical resource blocks (PRBs) in the configuration, and T_s^{μ} is the slot duration defined in (4). Both depend on the chosen numerology μ.

$$\text{DL (Mbps)} = 10^{-6} \cdot \left(8 \cdot \frac{946}{1024} \cdot \frac{N_{\text{PRB}}^{\text{BW},\mu} \cdot 12}{T_s^{\mu}} \cdot (1 - 0.14) \right) \cdot \left(\frac{N_{\text{sym}}^{\text{DL}}}{N_{\text{sym}}^{\text{Total}}} \right) \tag{2}$$

$$\text{UL (Mbps)} = 10^{-6} \cdot \left(8 \cdot \frac{946}{1024} \cdot \frac{N_{\text{PRB}}^{\text{BW},\mu} \cdot 12}{T_s^{\mu}} \cdot (1 - 0.08) \right) \cdot \left(\frac{N_{\text{sym}}^{\text{UL}}}{N_{\text{sym}}^{\text{Total}}} \right) \tag{3}$$

$$T_s^{\mu} = \frac{10^{-3}}{14 \cdot 2^{\mu}} \tag{4}$$

IV. Testbed Setup

Because the testbed is designed to evaluate the LAVP use case, it must support testing with multiple UEs simultaneously. Following the approach in [3], we adopted a coaxial cable and RF power splitter configuration. This provides an RF channel with known and stable characteristics, enabling repeatable and comparable measurements. By introducing an additional four-way splitter, the setup can accommodate four UEs while also connecting a spectrum analyser. The complete RF configuration is shown in Fig. 2, where the SDR is connected to a network of RF power splitters, linking four modems and the analyser to the RAN.

The software environment is distributed across three machines. The first runs the OAI software stack (core and gNB) on Ubuntu, connected to the SDR via USB 3.0. A second machine manages the four modems, each assigned to a separate virtual machine (VM) running Ubuntu Server. This provides isolated control of each UE through the Quectel QConnect software. The VMs are hosted on Proxmox, a type-1 hypervisor, with the USB ports of each modem passed directly to its respective VM. Automation is provided by a third orchestration machine, which runs Ansible scripts to coordinate all machines and VMs. This improves reproducibility and enables efficient execution of multiple test iterations. The orchestration machine and all testbed nodes are connected via a dedicated gigabit Ethernet management network. Fig. 3 illustrates the distribution of roles across the machines.

The hardware and software specifications of all components are listed in Tables I and II, respectively.

TABLE I
Hardware specifications

Component	Specification
Machines	
CPU	Intel Core i9 (14900KF)
Cores	24 (8 HP cores, 16 HE cores)
RAM	32 GB (DDR5 5200 MHz)
Storage	2 TB (SATA SSD)
RF Components	
Radio unit	Ettus USRP B210 SDR
Spectrum Analyser	TekTronix RSA306B
Coaxial	Mini-Circuits Hand-Flex 141
Attenuators	Mini-Circuits VAT10A+
Splitters 4-Way	Mini-Circuits ZN4PD1-63-S+
Splitter 2-Way	Mini-Circuits ZN2PD1-63-S+
Splitter 2-Way (spectrum analyser)	Mini-Circuits ZX10R-14-S+
Synchronization source	Ettus OctoClock-G CDA 2990
UE radio interface	Quectel RM500Q-GL 5G Modem
SIM card	Sysmocom sysmoISIM-SJA5-9FV

V. Methodology

Six different RAN configurations, varying in bandwidth and slot allocation, were evaluated on our testbed. All configurations used channel N78 with a 30 kHz SCS. Table III lists the bandwidth and slot configurations tested. These were chosen to align with a subset of the results in [10], restricted to 1:1 configurations. Some configurations, however, failed on our testbed: when uplink traffic was initiated, the UE disconnected, with logs indicating that it had received an invalid timing advance offset. To mitigate these failures, the guard time during the transition from downlink to uplink was

TABLE II
SOFTWARE VERSION INFORMATION

Software	Version
Core & gNB Machine	
Ubuntu	24.04.2 LTS
Linux kernel	6.14.0-29-generic
OAI	v2.3.0 & integration 2025.w37
UHD driver	UHD_4.7.0.HEAD-0-ga5ed1872
UE VM Machine	
Proxmox	pve-manager/7.4-3/9002ab8a
Kernel	5.15.102-1-pve
UE VMs	
Ubuntu Server	24.04.2 LTS
Linux kernel	6.14.0-29-generic
5G modem software	Quectel Qconnect Manager V1.6.7
iperf3	iperf 3.16 (cJSON 1.7.15)
Orchestration Machine	
Ubuntu	24.04.2 LTS
Linux kernel	6.14.0-29-generic
Ansible	core 2.18.6

widened from 2 symbols to 6 symbols. The modified values are marked with an asterisk.

A. Experiment Flow

The primary key performance indicators (KPIs) measured were uplink and downlink throughput, supplemented with spectrum analyser readings for additional insight.

For each configuration, the 5G core and RAN were restarted with the new parameters. Once the UE successfully attached to the network, downlink and uplink throughput were measured. Each throughput test lasted 10 seconds, with results recorded once per second. A 10 second idle period was included between tests to allow the network to stabilise. All results were logged to text files for subsequent processing. Each configuration was tested in 10 independent runs to ensure repeatability.

B. Testing Tools

Although the automation framework is outside the scope of this paper, here we describe the setup of `iperf3`, which was used for throughput measurement. Table II lists the software versions.

1) Iperf3 Server Setup: An `iperf3` server listening on port 5000 was started inside the "oai-ext-dn" container of the OAI core with the following command:

```
docker exec -itd oai-ext-dn bash -c
 ↪  "iperf3 -s -p 5000"
```

2) Iperf3 Client Setup: As in [10], User Datagram Protocol (UDP) traffic was chosen to avoid flow-control effects that could bias 5G scheduling. `iperf3` was configured to transmit at 120 Mbps, corresponding to the maximum theoretical throughput from 3GPP calculations (see Table IV).

The client ran on the VM connected to the 5G modem, using the following commands. Here, "192.168.70.135" is the IP address of the "oai-ext-dn" container, "-R" reverses the traffic direction for downlink, and "-t" specifies the test duration.

```
# Uplink
iperf3 --udp --bitrate 120M -c
 ↪  192.168.70.135 -p 5000 -t 10

# Downlink
iperf3 --udp --bitrate 120M -c
 ↪  192.168.70.135 -R -p 5000 -t 10
```

3) Ping Command: To test the round-trip time (RTT) latency, the ping command is set up to run 10 times at an interval of 1 second. The following command was used, where "10.0.0.1" is the default gateway of the UE to the 5G network.

```
  ping 10.0.0.1 -c 10
```

With this configuration, the RTT latency is measured from the UE VM to the User Plane Function (UPF) container in the OAI core. The related work used to compare the measurements against does not measure their latency with the UPF, instead opting to host the UE and gNB on the same machine and measuring the time difference between ingress and egress traffic to compute one-way delay statistics. Within our measurement setup the use of the UPF as an endpoint should not introduce significant latency since it is hosted on the same machine as the gNB.

VI. EXPERIMENTAL RESULTS USING OAI RAN V2.3.0

Based on the methodology described in Section V, we first evaluated the six RAN configurations using the current release version of the OAI RAN, namely v2.3.0.

A. Failures

Two of the six configurations consistently failed to run, even with the guard time modification described in Section V. Both failing configurations used a bandwidth of 20 MHz with TDD periodicities of 10 and 20 slots, respectively. These configurations were therefore excluded from the result figures for clarity.

B. Results

Figures 4a and 4b present the downlink and uplink throughput measurements, with corresponding numerical values listed in Table IV alongside latency measurements. The plots include the theoretical throughput values calculated using (2) and (3), as well as results from [10]. Note that in [10], numerical data was provided only for the 20 MHz cases; the 40 MHz means and latencies were extracted from graphs and are accurate to within 2.5 Mbps and 1 ms respectively, given the grid spacing. Seeing that the latency measurements in [10] were performed as a one-way latency measurement, the uplink and downlink latency measurements were summed to compare them with the RTT measurements made in this study.

Comparison between measurement means in Table IV are calculated as a percentage difference. Since the comparison

TABLE III
BANDWIDTH AND SLOT CONFIGURATION DIFFERENCES TESTED

Bandwidth (MHz)	PRBs	TDD Periodicity (slots)	DL Slots	DL Symbols	UL Slots	UL Symbols	Status
20	51	5	2	4*	2	4*	Modified
20	51	10	5	4*	4	4*	Fail
20	51	20	10	4*	9	4*	Fail
40	106	5	2	6	2	6	Success
40	106	10	5	4*	4	6*	Modified
40	106	20	10	4*	9	4*	Modified

is between measured values, the percentage difference is calculated as the difference of the values divided by their average as seen in (5).

$$\% \text{ Diff.} = \left| \frac{\text{Val}_1 - \text{Val}_2}{(\text{Val}_1 + \text{Val}_2)/2} \right| \times 100 \tag{5}$$

In the box plots, the spread of measured throughput is shown in blue, where the solid lines indicate the quartiles and median, and the dashed line indicates the mean. The red line represents the theoretical maximum throughput, while the green line shows the reference values from [10].

C. Discussion

1) Comparison with related work: As shown in Fig. 4a, the downlink means correlate strongly with the results from [10], with differences under 3% for all configurations except "10-slot-40MHz," which failed in [10]. A similar trend is seen in the uplink (Fig. 4b), except for the "5-slot-20MHz" case, which shows a significant difference of 31.37%. When it comes to latency there is a significant difference of up to 57.14% between the measured results and [10]. The difference in latency may be due in part to the different measurement points used in our study, but it is unlikely to be the primary cause of such a large difference.

2) Mean vs. median differences: In Fig. 4a, the mean is consistently about 10 Mbps lower than the median. This is explained by outliers that reduce the mean. Fig. 5a shows that the outliers occur during the first two seconds of each test, before the network reaches a steady state. By contrast, Fig. 5b shows uplink throughput remains relatively stable throughout.

3) Comparison with theoretical maximum: Figures 4a and 4b also highlight clear differences relative to the theoretical maximum throughput. Downlink results follow theoretical values closely, with only a small offset, while uplink throughput is substantially lower. The gap between measured and theoretical uplink throughput widens further with longer TDD periodicities. These findings motivated a closer inspection of the RF signals using a spectrum analyser.

4) Spectrum analyser investigation: Examining the RF output of the gNB revealed details of the 5G frame structure. In Fig. 6, relevant components are highlighted: green indicates the synchronisation signal block (SSB), orange shows full uplink slots, purple indicates uplink portions of the special slots, and blue spikes represent the physical downlink control channel (PDCCH) carrying downlink control information (DCI).

Two issues are immediately visible. First, uplink power levels appear low relative to the noise floor, DCI, and SSB. Second, in the 10-slot and 20-slot configurations, only two uplink slots are used out of the expected four and nine. Investigation of the OAI GitLab issues page confirmed this behaviour: a scheduling bug prevented utilisation of all uplink slots. A corresponding fix was identified in a merge request [12]. Consequently, we updated to the latest development branch, which included the fix, and repeated the measurements. These updated results are reported in Section VII. Unfortunately, there are no visible explanations for the difference in latency.

VII. EXPERIMENTAL RESULTS USING OAI RAN *2025.w37*

Building on the results of Test 1, we repeated the experiments using the latest weekly integration of the OAI RAN development branch available at the time of writing (integration 2025.w37). The experimental procedure remained identical to Test 1, with the only difference being the RAN software version.

A. Failures

As in Test 1, both 20 MHz configurations with TDD periodicities of 10 and 20 slots failed. In addition, the "20-slot-40MHz" configuration now failed at startup, producing the assertion shown in Fig. 7. The error message indicated that the slot configuration was incompatible with the current implementation, suggesting significant changes in supported TDD patterns. All failed configurations were excluded from the result figures for clarity.

B. Results

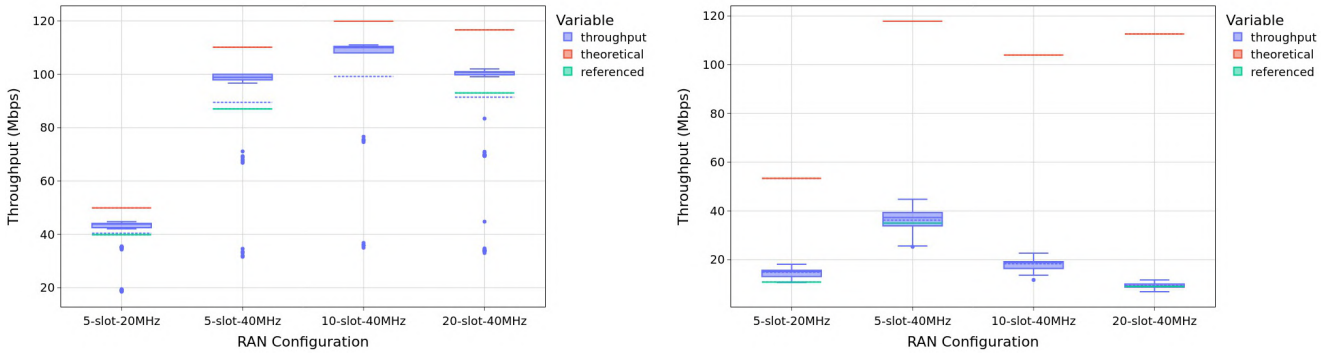
Figures 8a and 8b present the downlink and uplink throughput results, with the corresponding numerical values listed in Table V alongside latency measurements. The plots again include the theoretical maximum throughput calculated using (1). Unlike Test 1, the comparison here is made against the mean results from Test 1, with the percentage difference shown in the final column of Table V. The box plots follow the same structure described in Section VI-B.

C. Discussion

1) Downlink: Downlink performance in Test 2 is nearly identical to Test 1, with differences statistically insignificant at the 95% confidence interval.

TABLE IV
TEST 1 MEASUREMENT RESULTS COMPARED AGAINST THEORETICAL MAXIMUM AND RELATED WORK.

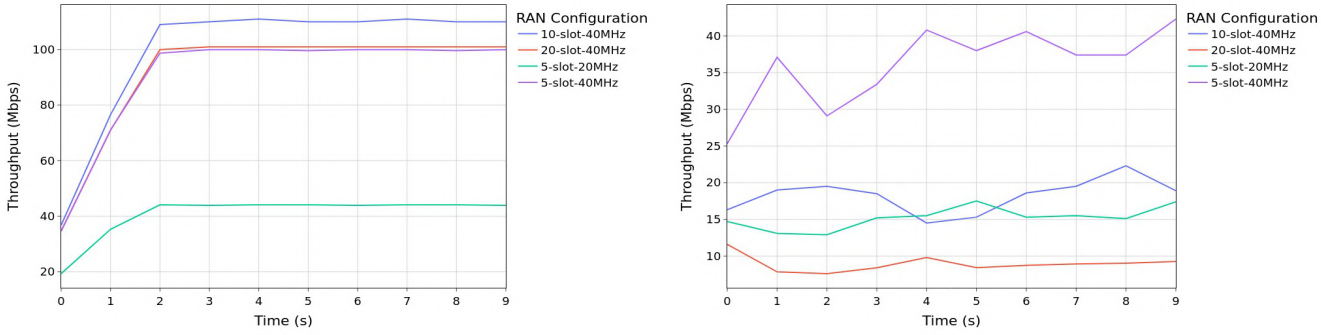
Measurement Type	RAN Configuration	Median	Mean	Std. Dev.	Theoretical Max	Ref. Mean [10]	% Diff.
DL Throughput (Mbps)	5-slot-20MHz	43.9	40.40	7.66	49.90	**39.90** (avg. of 3 tests)	1.25
	10-slot-20MHz	-	-	-	-	-	-
	20-slot-20MHz	-	-	-	-	-	-
	5-slot-40MHz	98.9	89.49	21.17	110.19	87 (from graph)	2.82
	10-slot-40MHz	110.0	99.18	23.64	119.91	0 (failed)	-
	20-slot-40MHz	101.0	91.45	20.99	116.67	93 (from graph)	1.68
UL Throughput (Mbps)	5-slot-20MHz	15.1	14.86	1.87	53.38	**10.83** (avg. of 3 tests)	31.37
	10-slot-20MHz	-	-	-	-	-	-
	20-slot-20MHz	-	-	-	-	-	-
	5-slot-40MHz	37.3	36.24	4.58	117.87	35 (from graph)	3.48
	10-slot-40MHz	18.8	18.32	2.21	104.01	0 (failed)	-
	20-slot-40MHz	9.12	9.36	1.13	112.67	9 (from graph)	3.92
RTT Latency (ms)	5-slot-20MHz	20.1	19.8	2.08	-	7 + 4 = 11 (from graph)	57.14
	10-slot-20MHz	-	-	-	-	-	-
	20-slot-20MHz	-	-	-	-	-	-
	5-slot-40MHz	19.8	19.7	1.96	-	7 + 4 = 11 (from graph)	56.68
	10-slot-40MHz	19.7	19.4	2.02	-	0 (failed)	-
	20-slot-40MHz	33.2	33.0	6.25	-	5 + 21 = 26 (from graph)	23.72



(a) Downlink Throughput (Mbps)

(b) Uplink Throughput (Mbps)

Fig. 4. Box plots of downlink and uplink throughput. Solid lines indicate quartiles and median; dashed lines indicate the mean.



(a) Downlink Time Series (Mbps)

(b) Uplink Time Series (Mbps)

Fig. 5. Example time series of downlink and uplink throughput from one test iteration.

TABLE V
TEST 2 MEASUREMENT RESULTS COMPARED AGAINST THEORETICAL MAXIMUM AND TEST 1 RESULTS.

Measurement Type	RAN Configuration	Median	Mean	Std. Dev.	Theoretical Max	Test 1 Mean	% Diff.
DL Throughput (Mbps)	5-slot-20MHz	44.0	40.77	7.51	49.90	40.40	0.91
	10-slot-20MHz	-	-	-	-	-	-
	20-slot-20MHz	-	-	-	-	-	-
	5-slot-40MHz	99.7	89.82	21.20	110.19	89.49	0.37
	10-slot-40MHz	110.0	98.87	24.04	119.91	99.18	0.31
	20-slot-40MHz	-	-	-	-	91.45	-
UL Throughput (Mbps)	5-slot-20MHz	15.3	15.17	1.84	53.38	14.86	2.06
	10-slot-20MHz	-	-	-	-	-	-
	20-slot-20MHz	-	-	-	-	-	-
	5-slot-40MHz	37.2	36.06	4.42	117.87	36.24	0.50
	10-slot-40MHz	34.1	**34.38**	4.15	104.01	**18.32**	**+60.95**
	20-slot-40MHz	-	-	-	-	9.36	-
RTT Latency (ms)	5-slot-20MHz	20.2	21.3	8.28	-	19.8	7.30
	10-slot-20MHz	-	-	-	-	-	-
	20-slot-20MHz	-	-	-	-	-	-
	5-slot-40MHz	19.7	19.8	1.77	-	19.7	0.51
	10-slot-40MHz	19.8	19.7	2.05	-	19.4	1.53
	20-slot-40MHz	-	-	-	-	33.0	-

TABLE VI
FORCED UPLINK SCHEDULING LATENCY RESULTS COMPARED AGAINST RELATED WORK.

Measurement Type	RAN Configuration	Median	Mean	Std. Dev.	Ref. Mean [10]	% Diff.
RTT Latency (ms)	5-slot-20MHz	8.4	8.9	1.81	7 + 4 = 11 (from graph)	21.11
	10-slot-20MHz	-	-	-	-	-
	20-slot-20MHz	-	-	-	-	-
	5-slot-40MHz	8.3	8.5	1.41	7 + 4 = 11 (from graph)	25.68
	10-slot-40MHz	10.6	11.5	2.91	0 (failed)	-
	20-slot-40MHz	-	-	-	5 + 21 = 26 (from graph)	-

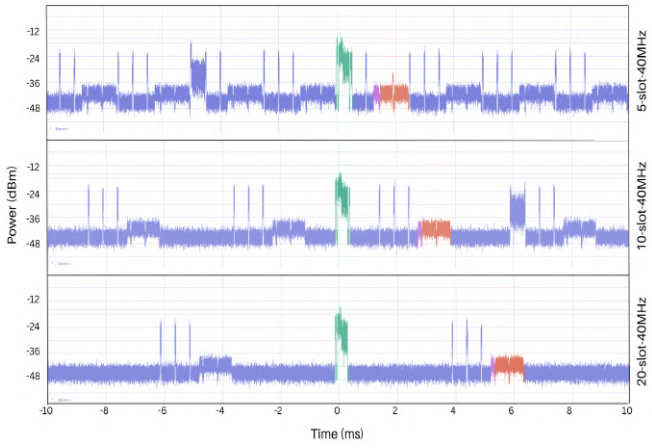


Fig. 6. RAN v2.3.0 spectrum analyser time-domain signals with uplink slots highlighted in red.

```
Assertion (tda_list->list.count <= 16) failed!
In nr_rrc_config_ul_tda() /home/oai/openairinterface5g/openair2/LAYER2/NR_MAC_gNB/nr_radio_config.c:1163
cannot reach all UL slots with current TDD configuration
```

Fig. 7. OAI RAN log output for failed "20-slot-40MHz" configuration in integration 2025.w37.

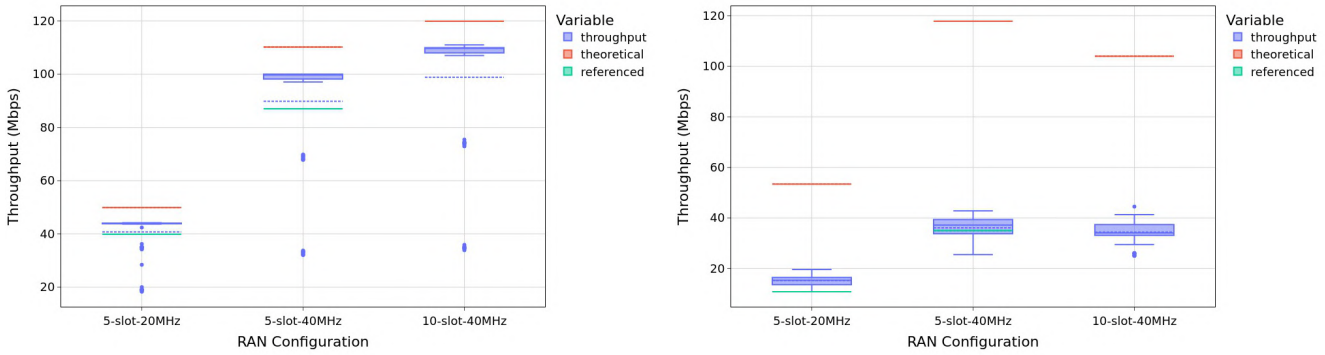
2) Latency: Latency performance remains similar to Test 1, with only the "5-slot-20MHz" configuration further increasing by 7.30%. The newer software version did not manage to decrease the latency to a value comparable to [10]. This could be explained by a RAN configuration difference between the two testbeds.

By limiting the maximum frame inactivity of the uplink shared channel (ULSCH) in the RAN configuration with the same testing procedure, the latency improved to results comparable to [10]. The parameter used to achieve this was added in the gNB configuration file as follows:

```
MACRLCs = ({
  ulsch_max_frame_inactivity = 0;
});
```

The configuration reduces latency by eliminating the need for UEs to send scheduling requests when data is available to send. The gNB automatically schedules uplink grants for every frame allowing the UE to send data more often since it can skip the scheduling request, greatly reducing latency. It is unclear whether this configuration parameter was set within the related work. However, it results in values much closer than before, but now lower by a margin of up to 25.68%. The discrepancy between latency results emphasises the need for open sharing of configurations and results to ensure reproducibility and use within future work. The results and comparison can be seen in Table VI.

3) Impact of TDD periodicity on uplink: Figure 8b shows that the large discrepancy between the 5-slot and 10-slot periodicities observed in Test 1 has disappeared. Spectrum analysis (Fig. 9) confirms that the 10-slot configuration now uses all four available uplink slots. This is reflected in Table V, where the "10-slot-40MHz"



(a) Downlink Throughput (Mbps)

(b) Uplink Throughput (Mbps)

Fig. 8. Box plots of downlink and uplink throughput for Test 2. Solid lines show quartiles and median; dashed lines show the mean.

uplink throughput improved by 60.95% relative to Test 1. However, this improvement coincided with the failure of the "20-slot-40MHz" configuration.

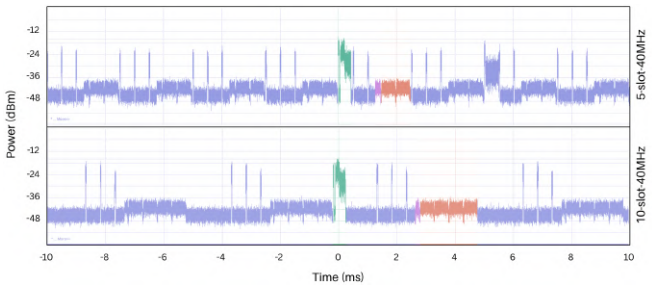


Fig. 9. RAN 2025.w37 spectrum analyser traces for slot configurations. Uplink slots are highlighted in red.

4) Comparison with theoretical maximum: Despite the improvement, Fig. 8b shows that uplink throughput still falls well below theoretical maximums. Low uplink power levels remain visible in Fig. 9, initially suggesting power limitation at the UE. However, the periodic media access control (MAC) statistics logs (Fig. 10) disprove this.

```
[NR_MAC]   Frame.Slot 768.0
UE RNTI 9bec CU-UE-ID 1 in-sync PH 58 dB PCMAX 23 dBm, average RSRP -71 (32 meas)
UE 9bec: UL-RI 1, TPMI 0
UE 9bec: dlsch_rounds 22900/0/0/0, dlsch_errors 0, pucch0_DTX 0, BLER 0.00000 MCS (1) 26
UE 9bec: ulsch_rounds 10056/698/0/0, ulsch_errors 0, ulsch_DTX 1, BLER 0.13434 MCS (1) 12 (Qm 6 deltaMCS 0 dB) NPRB 51 SNR 15.0 dB
UE 9bec: MAC:    TX       102939154 RX       11799882 bytes
UE 9bec: LCID 1: TX            1002 RX            3815 bytes
UE 9bec: LCID 2: TX               0 RX               0 bytes
UE 9bec: LCID 4: TX       102551943 RX       11631751 bytes
```

Fig. 10. Excerpt from periodic RAN MAC statistics showing high BLER.

Highlighted in Fig. 10 are three critical indicators: (1) power headroom (PH) (blue), which shows the UE still has headroom to increase transmit power; (2) signal-to-noise ratio (SNR) (green), which is adequate at the gNB; and (3) BLER and modulation coding scheme (MCS) (red). The UE operated at MCS 12, far below the maximum of 27 corresponding to 256-quadrature amplitude modulation (QAM). Because the RAN adjusts MCS according to BLER, and the observed BLER remained above the 0.05 threshold (13% errors), the scheduler could not increase the coding rate. This indicates that uplink decoding remains unreliable even at relatively low MCS.

Low uplink throughput is seen in multiple related works, all with different theories to its cause. In [9], the throughput shortfall is attributed to the fact that OAI is still in development and that uplink (UL) is not yet fully implemented; however, this study was tested on a considerably older version of OAI from 2023. In [13], the authors performed similar tests using srsRAN instead of OAI, in which they attributed the poor uplink performance to processing limitations in their workstations, arising from their general-purpose nature. They also mentioned that parts of the SDR's RF chain could have introduced noise. Alternatively, [14] believed the problem occurred because of resource allocation problems in srsRAN. However, they noted that both SDRs used in testing (USRP B210 and X410) exhibited poor uplink performance, but that the X410 did so to a lesser degree.

In evaluating these hypotheses, the study by [13] provides the most compelling argument. The observed poor uplink performance across both OAI and srsRAN suggests that issues related to slow development or resource allocation are improbable. The superior performance of the USRP X410 compared to the B210 aligns with the hypothesis that noise within the SDR RF chain could be a factor, given that the X410 is a more premium model. Alternatively, the performance difference may be attributed to the more efficient Ethernet-based front-haul interface of the X410 versus the USB interface of the B210.

A possible remedy is to increase the target SNR requested by the gNB, thereby forcing the UE to transmit at higher power. Preliminary experiments with higher target SNRs confirm that MCS increases, but link stability deteriorates significantly once the SNR target is set above 20 dB. We believe this is because the UE power output exceeds the SDR maximum input limit, resulting in clipping. This trade-off requires further investigation. The source of the problem could be narrowed down by testing the same setup with a commercial

Open-RAN radio unit (O-RU), but we currently do not have access to such an RU.

VIII. Conclusion

This work evaluated the suitability of OAI-based 5G testbeds for uplink-heavy applications such as LAVP. By benchmarking six RAN configurations against both 3GPP theoretical throughput calculations and related work, we demonstrated that our testbed produces results consistent with prior studies for downlink performance. However, uplink throughput consistently fell short of theoretical expectations. Poor uplink performance severely affects LAVP applications because video demands high bandwidth and stable network conditions to deliver optimal quality of experience (QoE) for viewers.

Through spectrum analysis and log inspection, we identified a scheduler bug in OAI v2.3.0 that prevented full utilisation of uplink slots, particularly for longer TDD periodicities. After updating to integration 2025.w37, we confirmed that this issue was resolved, with the "10-slot-40MHz" configuration showing a 60.95% improvement in uplink throughput. Despite this, uplink performance remains well below theoretical potential, with high BLER preventing the use of higher MCS values. Future work may look into analysing PRB utilisation during UL slot scheduling by leveraging FlexRIC, to determine whether all available PRBs are fully allocated within the OAI RAN.

Our contributions are threefold:

- We provide a validated and reproducible testbed for OAI-based 5G performance studies, including detailed configuration files and raw measurement data as a GitHub repository [15].
- We identified, validated, and confirmed the fix of a critical uplink scheduling issue in OAI.
- We highlight remaining challenges in achieving theoretical uplink performance, underscoring the need for further investigation into MCS selection, target SNR tuning, and power control.

These results reinforce the importance of validating testbeds against both theoretical models and related work, while also emphasising the sensitivity of results to specific software versions and hardware configurations. Although continued development on OAI can result in the measured results quickly becoming outdated, the work provided us with an automated platform to validate the testbed with newer software revisions. We encourage continued experimentation with OAI, the reporting of issues, and the open sharing of reproducible measurement data to accelerate progress toward reliable 5G testbeds for uplink-heavy use cases such as LAVP.

References

[1] K. Rogers. (2024, Sep) livestreaming. Encyclopedia Britannica. [Online]. Available: https://www.britannica.com/technology/livestreaming

[2] (2024, Sep) 5G-RECORDS Project. [Online]. Available: https://www.5g-records.eu/

[3] B. Agarwal, C. Felber, N. Pandeya, F. Kaltenberger, M. Ruffini, and G.-M. Muntean, "Analysis of Real-Time Video Streaming and Throughput Performance Using the Open Air Interface Stack on Multiple UEs," in *2023 IEEE Conference on Standards for Communications and Networking (CSCN)*. Munich, Germany: IEEE, Nov. 2023, pp. 235–240. [Online]. Available: https://ieeexplore.ieee.org/document/10453217/

[4] E.-Z. G. Bozis, N. C. Sagias, M. C. Batistatos, M.-A. Kourtis, G. K. Xilouris, and A. Kourtis, "A Versatile 5G Standalone Testbed Based On Commodity Hardware," in *2024 Panhellenic Conference on Electronics & Telecommunications (PACET)*. Thessaloniki, Greece: IEEE, Mar. 2024, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/10497086/

[5] M. Chepkoech, E.-R. Modroiu, J. Mwangama, M. Corici, and T. Magedanz, "Evaluation of OSS-Enabled OpenRAN Compliant 5G StandAlone Campus Networks," in *2023 International Conference on Electrical, Computer and Energy Technologies (ICECET)*. Cape Town, South Africa: IEEE, Nov. 2023, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/10389574/

[6] T. O. Atalay, A. Famili, D. Stojadinovic, and A. Stavrou, "Demystifying 5G Traffic Patterns with an Indoor RAN Measurement Campaign," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*. Kuala Lumpur, Malaysia: IEEE, Dec. 2023, pp. 1185–1190. [Online]. Available: https://ieeexplore.ieee.org/document/10437330/

[7] M. Seidel, A. I. Grohmann, P. Sossalla, F. Kaltenberger, and F. H. Fitzek, "How to Get Away with OpenAirInterface: A practical Guide to 5G RAN Configuration," in *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. Tenerife, Canary Islands, Spain: IEEE, Jul. 2023, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10252787/

[8] F. John, J. Schuljak, L. B. Vosteen, B. Sievers, A. Hanemann, and H. Hellbruck, "A Reference Deployment of a Minimal Open-Source Private Industry and Campus 5G Standalone (SA) System," in *2022 IEEE 10th International Conference on Information, Communication and Networks (ICICN)*. Zhangye, China: IEEE, Aug. 2022, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/10006563/

[9] A. Sahbafard, R. Schmidt, F. Kaltenberger, A. Springer, and H.-P. Bernhard, "On the Performance of an Indoor Open-Source 5G Standalone Deployment," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*. Glasgow, United Kingdom: IEEE, Mar. 2023, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10118776/

[10] S. Raffeck, S. G. Grøsvik, S. Lange, T. Hossfeld, T. Zinner, and S. Geissler, "Parameterizing 5G New Radio: A Comparative Measurement Study on Throughput and Delay," in *2024 20th International Conference on Network and Service Management (CNSM)*. Prague, Czech Republic: IEEE, Oct. 2024, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/10814615/

[11] 3rd Generation Partnership Project, "User Equipment (UE) radio access capabilities (3GPP TS 38.306 version 18.6.0 Release 18)," 3GPP, TS 38.306, July 2025.

[12] R. Schmidt. (2025, Jul) Refactor ULSCH scheduler towards more TDD pattern flexibility. OpenAirInterface GitLab Merge Request 3521. [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/-/merge_requests/3521

[13] J. E. Håkegård, H. Lundkvist, A. Rauniyar, and P. Morris, "Performance Evaluation of an Open Source Implementation of a 5G Standalone Platform," *IEEE Access*, vol. 12, pp. 25 809–25 819, 2024.

[14] M. Amini, A. El-Ashmawy, C. Rosenberg, and A. K. Khandani, "5G DIY: Impact of Different Elements on the Performance of an E2E 5G Standalone Testbed," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*. Kuala Lumpur, Malaysia: IEEE, Dec. 2023, pp. 6377–6382.

[15] A. Oelofse. (2025) Validation of an OAI based 5G testbed. [Online]. Available: https://github.com/An3sOelofse/Validation-of-an-OAI-based-5G-testbed.git