مركز الأكاديمية للنشر
ACADEMY Publishing Center

ACE

ADVANCES in COMPUTING & ENGINEERING

JOURNAL

VOLUME 6
ISSUE 1
JUNE 2026
E-ISSN: 2735-5985

Academy Publishing Center
Journal of Advances in Computing and Engineering (ACE) First edition 2021

Arab Academy for Science, Technology, and Maritime Transport, AASTMT Abu Kir Campus, Alexandria, EGYPT
P.O. Box: Miami 1029
Tel: (+203) 5622366/88 – EXT 1069 and (+203) 5611818
Fax: (+203) 5611818
Web Site: http://apc.aast.edu

No responsibility is assumed by the publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

Every effort has been made to trace the permission holders of figures and in obtaining permissions where necessary.

Volume 6, Issue 1, June 2026
ISSN: 2735-5977 Print Version
ISSN: 2735-5985 Online Version

ACE
Journal of Advances in Computing and Engineering

Journal of Advances in Computing and Engineering (ACE) is a peer-reviewed, open access, interdisciplinary journal with two issues yearly. The focus of the journal is on theories, methods, and applications in computing and engineering. The journal covers all areas of computing, engineering, and technology, including interdisciplinary topics.
All submitted articles should report original, previously unpublished research results, experimental or theoretical. Articles submitted to the journal should meet these criteria and must not be under consideration for publication elsewhere. The manuscript should follow the style specified by the journal.
ACE also publishes survey and review articles in the scope. The scope of ACE includes but not limited to the following topics: Computer science theory; Algorithms; Intelligent computing; Bioinformatics; Health informatics; Deep learning; Networks; Wireless communication systems; Signal processing; Robotics; Optical design engineering; Sensors.

The journal is open-access with a liberal Creative Commons Attribution-NonCommercial 4.0 International License which preserves the copyrights of published materials to the authors and protects it from unauthorized commercial use.
The ACE journal does not collect Articles Processing Charges (APCs) or submission fees, and is free of charge for authors and readers, and operates an online submission with the peer review system allowing authors to submit articles online and track their progress via its web interface. The journal is financially supported by the Arab Academy for Science, Technology, and Maritime Transport AASTMT in order to maintain quality open-access source of research papers on Computing and Engineering.

ACE has an outstanding editorial and advisory board of eminent scientists, researchers and engineers who contribute and enrich the journal with their vast experience in different fields of interest to the journal.

Editorial Committee

EDITOR IN CHIEF

Yousry Elgamel, Ph.D
Professor, Communication and Information Technology,
Arab Academy for Science, Technology and Maritime Transport (AASTMT)
Former Minister of Education, Egypt.
E-mail: yelgamal@aast.edu

ASSOCIATE EDITORS

Amira Ibrahim Zaki, Ph.D
Professor, Electronics and Communication Engineering
Arab Academy for Science and Technology and Maritime Transport (AASTMT)
Abu Kir Campus, POBox: 1029 Miami, Alexandria, EGYPT
Email: amzak10@aast.edu

Fahima Maghraby, Ph.D
Associate Professor, Computer Science
Arab Academy for Science, Technology, and Maritime Transport(AASTMT)
El-Moshier Ahmed Ismail, Postgraduates Studies Bldg, PO Box 2033-Elhorria, Cairo, EGYPT
Email: fahima@aast.edu

Hanady Hussien Issa, Ph.D
Professor, Electronics and Communication Engineering
Arab Academy for Science, Technology, and Maritime Transport (AASTMT)
El-Moshier Ahmed Ismail, Postgraduates Studies Bldg, PO Box 2033-Elhorria, Cairo, EGYPT
Email: hanady.issa@aast.edu

Nahla A. Belal, Ph.D
Professor, Computer Science,
Arab Academy for Science, Technology, and Maritime Transport (AASTMT)
Bldg. 2401 A, Smart Village, 6 October City, PO Box 12676, Giza, EGYPT
Email: nahlabelal@aast.edu

Editorial Board

Hussien Mouftah, Ph.D
Professor, School of Electrical Engineering and Computer Science University of Ottawa, Canada.

Michael Oudshoorn, Ph.D
Professor, School of Engineering, High Point University, United States.

Mudasser F. Wyne, Ph.D
Professor, Engineering and Computing Department, National University, USA.

Nicholas Mavengere, Ph.D
Senior Lecturer, Faculty of Science & Technology, Bournemouth University, UK.

Ramzi A. Haraty, Ph.D
Professor, Department of Computer Science and Mathematics, Lebanese American University, Lebanon

Sahra A Idwan, Ph.D
Professor, Department of Computer Science and Applications, Hashemite University, Jordan.

Advisory Board

Ahmed Abou Elfarag, Ph.D.
Professor, College of Artificial Intelligence, Arab Academy for Science and Technology and Maritime Transport (AASTMT), El-Alamein, Egypt

Ahmed Hassan, Ph.D
Professor, Dean of ITCS School, Nile University.

Aliaa Youssif, Ph.D
Professor, College of Computing and Information Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Smart Village, Egypt

Aly Fahmy, Ph.D
Professor, College of Artificial Intelligence, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Al-Alamein, Egypt

Amani Saad, Ph.D
Professor, College Of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Amr Elhelw, Ph.D
Professor, College Of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Cairo, Egypt

Attalah Hashad, Ph.D
Professor, College Of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), South Valley, Egypt

Atef Ghalwash, Ph.D
Professor, Faculty of Computers & Information, Chief Information Officer(CIO), Helwan University

Ayman Adel Abdel Hamid, Ph.D
Professor, College of Computing and Information Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Ehab Badran, Ph.D
Professor, College Of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Ismail Abdel Ghafar, Ph.D
President, Arab Academy for Science and Technology and Maritime Transport (AASTMT).

Khaled Ali Shehata, Ph.D
Professor, College Of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Cairo, Egypt

Khaled Mahar, Ph.D
Professor, College of Computing and Information Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Maha Sharkas, Ph.D
Professor, College Of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Meer Hamza, Ph.D
Professor, College of Computing and Information Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Mohamed Aboul Dahab , Ph.D
Professor, College Of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Cairo, Egypt

Mohamed Hashem, Ph.D
Professor, Faculty of Computer and Information Sciences-Ain shams University.

Moustafa Hussein, Ph.D
Professor, College of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Mohamed Essam Khedr, Ph.D
Professor, College of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Mohamed Kholief, Ph.D
Professor, College of Computing and Information Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Mohamed Shaheen, Ph.D
Professor, College of Computing and Information Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Nagwa Badr, Ph.D
Professor, Faculty of Computer and Information Sciences-Ain shams University.

Nashwa El-Bendary, Ph.D
Professor, College of Computing and Information Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), South Valley, Egypt

Osama Mohamed Badawy, Ph.D
Professor, College of Computing and Information Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Ossama Ismail, Ph.D
Professor, College of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Sherine Youssef, Ph.D
Professor, College of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Waleed Fakhr, Ph.D
Professor, College Of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Cairo, Egypt

Yasser El-Sonbaty, Ph.D
Professor, College of Computing and Information Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Yasser Hanafy, Ph.D
Professor, College of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt

Administrative Committee

JOURNAL MANAGER

Professor Yasser Gaber Dessouky, Ph.D
Professor, Dean of Scientific Research and Innovation,
Arab Academy for Science, Technology & Maritime Transport, Egypt.

COPY EDITOR & DOCUMENTATION MANAGER

Dina Mostafa El-Sawy,
Coordinator, Academy Publishing Center,
Arab Academy for Science , Technology & Maritime Transport, Egypt.

PUBLISHING CONSULTANT

Dr. Mahmoud Khalifa
Consultant, Academy Publishing Center,
Arab Academy for Science, Technology & Maritime Transport, Egypt.

LAYOUT EDITOR & PROOF-READER

Engineer Sara Gad,
Graphic Designer, Academy Publishing Center,
Arab Academy for Science, Technology & Maritime Transport, Egypt.

Engineer Yara El Rayess,
Graphic Designer, Academy Publishing Center,
Arab Academy for Science, Technology & Maritime Transport, Egypt.

IT MANAGER

Engineer Ahmad Abdel Latif Goudah,
Web Developer and System Engineer,
Arab Academy for Science, Technology & Maritime Transport, Egypt.

TABLE OF CONTENT

http://dx.doi.org/10.21622/ACE.2026.06.1.1704

CROSS-MODAL FEATURE LEARNING FOR MULTI-SENSOR IOT ANOMALY DETECTION: A COMPREHENSIVE EMPIRICAL ANALYSIS OF CYBER PHYSICAL SYSTEMS

Richard Chukwuebuka Nwachukwu

Ignatius Ajuru University of Education, Port Harcourt, Rivers State, Nigeria

Email: {iamrichardcn@gmail.com}

Received 28 September 2025 - Accepted 28 January 2026 - Published 02 April 2025

ABSTRACT

The proliferation of Internet of Things (IoT) devices has created unprecedented security challenges, with traditional single-modal anomaly detection systems proving inadequate against sophisticated multi-vector attacks. This paper presents a novel cross-modal feature learning framework that synergistically integrates environmental sensor telemetry with network traffic patterns for enhanced anomaly detection across heterogeneous IoT architectures. Through comprehensive experimentation on the TON_IoT dataset encompassing 380,609 synchronized records across three distinct IoT systems (weather monitoring, smart refrigeration, and GPS tracking), we demonstrate that cross-modal integration consistently outperforms single-modal approaches. Our Random Forest implementation achieved 95.35% accuracy for weather systems (0.50% improvement over sensor-only), 78.13% for refrigeration systems (37.34% improvement), and 95.24% for GPS systems (9.45% improvement). We also validated our approach using Support Vector Machines (93.87% average accuracy) and k-Nearest Neighbors (91.24% average accuracy), demonstrating the robustness of cross-modal integration across diverse algorithmic families. Feature importance analysis reveals system-specific optimization patterns: atmospheric pressure emerges as the primary discriminator in weather systems (19.8% importance), while network features dominate refrigeration systems (86.6% combined importance). Most significantly, we provide the quantitative evidence that 24.7% of anomalies manifest simultaneously across both sensor and network modalities, indicating sophisticated coordinated attacks that single-modal systems would partially miss. The proposed temporal alignment methodology successfully addresses heterogeneous timestamp formats and sampling rates, creating a reusable framework for cross-modal IoT security research. These findings establish cross-modal feature learning as essential for comprehensive IoT security, with practical implications for designing resilient cyber-physical systems.

Keywords: (IoT security, Cross-modal Learning, Anomaly Detection, Sensor Fusion, Cyber-Physical Systems, Machine Learning)

INTRODUCTION

The Internet of Things (IoT) paradigm has fundamentally transformed modern computing infrastructure, with deployment projections exceeding 75 billion connected devices by 2025 [1]. These cyber-physical systems generate massive volumes of heterogeneous data streams encompassing environmental measurements, network communications, and device telemetry, creating both unprecedented opportunities for intelligent monitoring and significant security vulnerabilities [2]. The inherent complexity of IoT ecosystems, characterized by resource-constrained devices, diverse communication protocols, and distributed architectures, presents unique challenges that traditional security approaches struggle to address effectively [3].

The landscape of IoT anomaly detection has evolved through distinct phases, each addressing specific limitations of previous approaches. Early IoT security systems

http://dx.doi.org/10.21622/ACE.2026.06.1.1704

employed statistical anomaly detection techniques such as Gaussian distribution models, Z-score anomaly detection, and statistical process control charts to identify deviations from normal behavior patterns [4]. These methods, while computationally efficient for resource-constrained devices, struggled with the high-dimensional and non-stationary nature of IoT data streams. The dynamic and heterogeneous characteristics of IoT environments often result in high false positive rates [5]. The application of supervised learning algorithms marked a significant advancement in detection capabilities. Ensemble methods, particularly Random Forests introduced by Breiman [6], have shown robust performance in handling mixed data types and noisy features common in IoT environments, with studies reporting 94-97% accuracy on network intrusion datasets [7]. Support Vector Machines (SVM) with radial basis function kernels have also demonstrated effectiveness in IoT anomaly detection, achieving 92-95% accuracy in industrial IoT applications [8]. However, these approaches require extensive labeled datasets, which are challenging to obtain due to the rarity and diversity of IoT attacks.

Unsupervised techniques have emerged as practical alternatives, with clustering-based methods such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) and k-means clustering, along with isolation forests, demonstrating effectiveness in identifying anomalies without prior attack knowledge [9]. Recent work by Xing and Wang (2025) introduced Cross-Modal Attention Networks using LSTM and Temporal Convolutional Networks (TCNs) for system software anomaly detection, achieving 12.3% F1-score improvements over single-modal baselines [10]. Regardless, this compartmentalized approach fundamentally fails to capture the intrinsic correlations between physical sensor measurements and network communication behaviors that characterize both normal operations and sophisticated attacks [6].

The limitations of single-modal approaches become particularly apparent when considering the evolution of IoT attack sophistication. Recent security incidents demonstrate that adversaries increasingly employ multi-stage attacks that begin with subtle sensor manipulations before escalating to network-level exploits [9]. For instance, the Stuxnet malware famously manipulated industrial sensor readings while simultaneously altering network communications to mask its presence [11]. Traditional detection systems analyzing only sensor data might miss network-based indicators, while network-focused approaches could overlook critical sensor manipulations that precede major breaches [12]. Multimodal fusion approaches have shown promise in related domains, with Ye (2025) reporting 92.70% F1-score using dynamic graph structures and deep feature fusion for multi-sensor networks [13], while Khan et al. (2025) achieved 92% ROC-AUC using multimodal autoencoder fusion for vehicle damage detection [14].

Despite significant advances in machine learning for IoT security [15], [16], existing research exhibits several critical limitations. Most studies have focused on enhancing detection within individual data modalities through techniques like deep autoencoders [10] and LSTM networks [11] rather than exploring synergistic cross-modal benefits [17]. The few attempts at multimodal integration often fail to address the fundamental challenge of aligning heterogeneous data streams with different timestamp formats and sampling rates [18]. Recent work by Li et al. (2025) demonstrated the value of multi-sensor fusion in industrial applications, achieving 96.1% AUROC by unifying RGB, laser scanner, and infrared thermography data [19], while Wang et al. (2024) proposed HybridCube for power transformer fault detection using cross-modal data fusion based on information entropy theory [20]. Existing approaches typically evaluate on single device types or specific application domains, lacking comprehensive validation across diverse IoT architectures.

A. MAIN CONTRIBUTIONS

This research makes the following key contributions to the field of IoT security and cross-modal anomaly detection:

1. The study introduces a comprehensive methodology for synchronizing heterogeneous IoT data streams with different timestamp formats (human-readable vs. Unix epoch)

http://dx.doi.org/10.21622/ACE.2026.06.1.1704

and varying sampling rates.

2. The study provides the first large-scale empirical evidence that 24.7% of IoT anomalies manifest simultaneously across both sensor and network modalities in real-world datasets.

3. The study demonstrates cross-modal integration effectiveness across five machine learning algorithms (Decision Trees, Gradient Boosting, Random Forest, Support Vector Machines, k-Nearest Neighbors), over single-modal approaches.

METHOD

AN EXPERIMENTAL DESIGN

The experimental framework employs a systematic five-phase approach designed to comprehensively evaluate cross-modal feature learning effectiveness across diverse IoT architectures. The methodology integrates data collection, preprocessing, feature engineering, model implementation, and performance evaluation through a structured pipeline that addresses the fundamental challenges of heterogeneous IoT data integration.

Figure 1 illustrates the complete methodology pipeline, demonstrating the sequential flow from raw data acquisition through synchronized cross-modal analysis. The pipeline architecture reveals the critical dependency between temporal alignment (Phase 2) and subsequent feature engineering (Phase 3), where proper synchronization of 380,609 records enables effective cross-modal correlation analysis. The validation checkpoints embedded throughout the pipeline ensure data integrity, with particular emphasis on maintaining temporal consistency across heterogeneous data streams from weather monitoring, smart refrigeration, and GPS tracking systems. The experimental design incorporates multiple comparison baselines to isolate the specific contributions of cross-modal integration, enabling quantitative assessment of performance improvements over single-modality approaches.

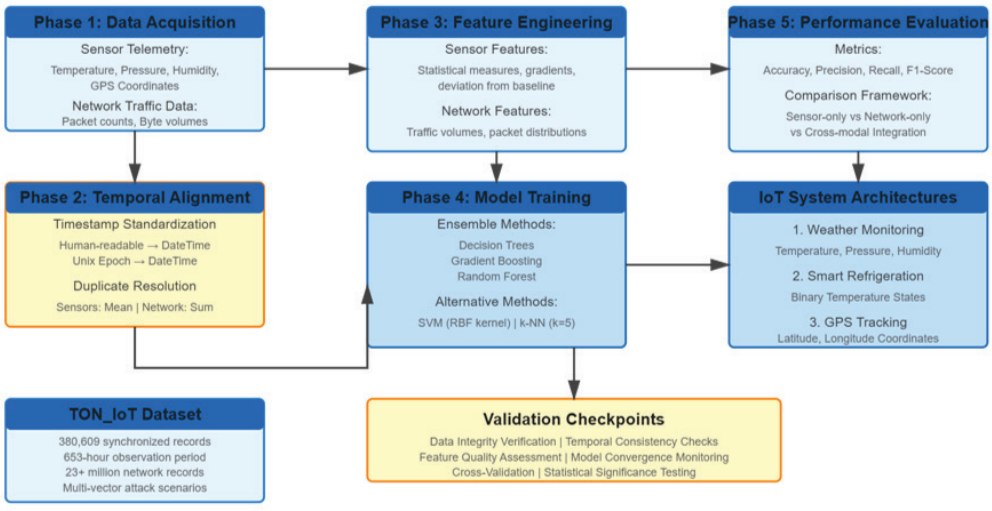


Figure 1: Cross-modal IoT anomaly detection methodology pipeline

Each phase includes validation checkpoints to verify data integrity and processing accuracy, with particular attention to maintaining temporal consistency across modalities. The design accommodates the inherent variability in IoT data characteristics while providing robust frameworks for comparative analysis across different system types and algorithm configurations.

B. DATASET DESCRIPTION AND CHARACTERISTICS

http://dx.doi.org/10.21622/ACE.2026.06.1.1704

The study utilizes the TON_IoT dataset, a comprehensive cybersecurity dataset specifically designed for IoT research applications [21]. This dataset provides extensive coverage of both benign operational patterns and diverse attack scenarios across multiple IoT device categories, establishing a realistic foundation for cross-modal security analysis. The dataset's comprehensive nature enables evaluation of detection approaches under varying operational conditions and threat landscapes.

The dataset encompasses three distinct IoT system architectures, each representing different application domains and sensor configurations. Weather monitoring systems incorporate environmental sensors measuring temperature, atmospheric pressure, and humidity levels, generating 650,242 sensor records alongside 22,339,021 network communication records over a 653-hour observation period. Smart refrigeration systems employ temperature monitoring with binary operational states, producing 347,162 sensor records with approximately 20 million network records across the same temporal window. GPS tracking systems capture geographical coordinates through latitude and longitude measurements, yielding 342,891 sensor records with corresponding network traffic volumes.

C. SAMPLING AND DATA INTEGRATION

The temporal alignment of heterogeneous data streams represents the study sample for this research. IoT systems generate data with fundamentally different temporal characteristics: sensor measurements typically employ human-readable timestamp formats ("31-Mar-19 12:36:52"), while network communications utilize Unix epoch timestamps (1556485532). This temporal heterogeneity necessitates comprehensive standardization procedures to enable cross-modal correlation analysis.

The standardization process employs pandas datetime conversion algorithms with explicit format specifications to ensure consistent temporal representation across data modalities. Timezone normalization procedures address potential geographic variations in data collection, while precision preservation techniques maintain sub-second accuracy where available. The standardization pipeline successfully processes over 23 million heterogeneous timestamp records while preserving temporal relationships essential for correlation analysis.

Duplicate timestamp resolution requires tailored aggregation strategies reflecting the distinct characteristics of different data types. Numeric Sensor Features - Mean aggregation preserves central tendency:

$$\check{S}(t) = \frac{1}{|\,S(t)\,|} \sum_{i=1}^{|S(t)|} s_i(t)$$

Network Traffic Volumes - Sum aggregation captures total activity:

$$\check{V}(t) = \sum_{j=1}^{|V(t)|} V_i(t)$$

Categorical Features - Mode selection preserves dominant characteristics:

$$\check{C}(\hat{}(t) = mode(C(t)) = arg^{max}_{c \in C(t)} |\,\{c' \in C(t) : c' = c\}\,|$$

This strategy successfully resolved 184,395 duplicate timestamps (53.1% of weather sensor data) while preserving essential characteristics.

D. COMPREHENSIVE DETECTION SYSTEM ARCHITECTURE

http://dx.doi.org/10.21622/ACE.2026.06.1.1704

Figure 2 presents the detailed architectural design of the proposed cross-modal anomaly detection system. The architecture consists of four primary layers. The four-layer architecture demonstrates clear separation of concerns, with each layer building upon the outputs of its predecessor. The data acquisition layer (Layer 1) handles the complexity of heterogeneous IoT protocols, successfully processing over 23 million network records alongside sensor telemetry. Layer 2's temporal alignment represents the critical innovation, resolving 184,395 duplicate timestamps through adaptive aggregation strategies. The feature engineering layer (Layer 3) extracts both modality-specific and cross-modal correlation features, with atmospheric pressure emerging as the most discriminative sensor feature (19.8% importance). Finally, the machine learning layer (Layer 4) demonstrates algorithm diversity. This layer includes model selection logic, hyperparameter optimization, and confidence scoring mechanisms for anomaly classification.

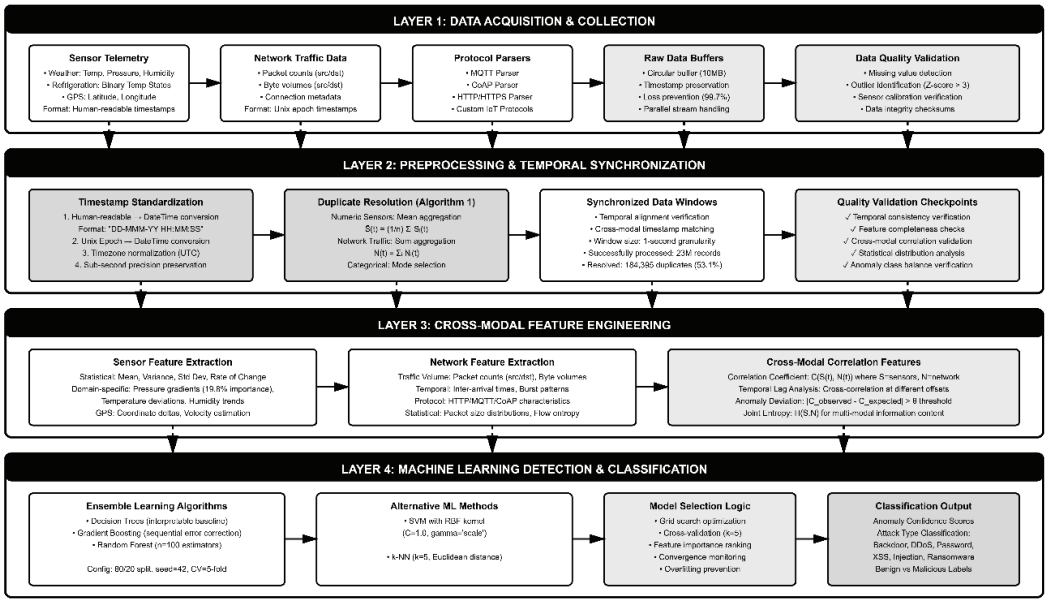


Figure 2: architectural design of the proposed cross-modal anomaly detection system.

E. MODEL DERIVATIONS

The theoretical foundation for cross-modal anomaly detection emerges from the fundamental cyber-physical coupling characteristics inherent in IoT systems. This coupling manifests through predictable relationships between environmental conditions, sensor responses, and subsequent network communications, creating correlation patterns that sophisticated attacks must inevitably disrupt. Figure 2 presents the conceptual framework for cross-modal IoT anomaly detection, illustrating the interconnected nature of physical sensor measurements and network traffic patterns. Cross-modal integration combines both data streams to achieve comprehensive anomaly detection, identifying coordinated attacks that would otherwise remain undetected by isolated approaches. The theoretical basis for cross-modal anomaly detection rests on the cyber-physical coupling inherent in IoT systems. Environmental changes trigger sensor readings according to physical laws:

$$S(t)=f_{\text{physical}}\,(E(t))+\epsilon_{\text{sensor}}(t)$$

where S(t) represents sensor measurements, E(t)denotes environmental conditions, and $\epsilon_{\text{sensor}}(t)$ captures measurement noise.

These sensor readings subsequently generate network communications:

$$N(t)=g_{\text{protocol}}\,(S(t),\Delta S(t))+\epsilon_{\text{network}}\,(t)$$

http://dx.doi.org/10.21622/ACE.2026.06.1.1704

where N(t) represents network traffic, ΔS(t) captures sensor state changes, and $\epsilon_{network}(t)$ models communication variability.

Under normal operations, these relationships exhibit predictable correlation patterns. Anomalies manifest as deviations from expected correlations:

$$A(t)=\|C_{observed}(S(t),N(t))-C_{expected}(S(t),N(t))\|>\theta$$

where C denotes correlation functions, and θ represents the anomaly threshold.

F. STATISTICAL ANALYSES

The experimental evaluation employs a comprehensive three-tier framework designed to isolate and quantify the specific contributions of cross-modal feature integration. This framework systematically compares three distinct analytical approaches: sensor-only baseline analysis utilizing exclusively environmental sensor measurements, network-only baseline analysis employing solely network traffic characteristics, and cross-modal integration combining both sensor and network features through synchronized temporal alignment. The sensor-only analysis establishes the discriminative capacity of environmental measurements independent of network characteristics, while the network-only analysis determines the detection capabilities of communication pattern analysis. Cross-modal integration performance can then be directly compared against both single-modality baselines to quantify the specific benefits of multimodal correlation analysis.

Algorithm selection encompasses five complementary machine learning approaches chosen for their distinct analytical strengths and widespread applicability in IoT security research. Decision Trees provide interpretable baseline performance with inherent feature selection capabilities, enabling direct identification of the most discriminative characteristics within each modality. Gradient Boosting implements sequential error correction through iterative model refinement, offering enhanced capability for capturing complex non-linear patterns in cross-modal relationships. Random Forest employs parallel ensemble learning with robust handling of mixed data types and missing values, characteristics particularly valuable in heterogeneous IoT environments. Support Vector Machines (SVM) with radial basis function (RBF) kernels provide strong performance for high-dimensional feature spaces through optimal hyperplane separation [8]. k-Nearest Neighbors (k-NN) offers instance-based learning that naturally captures local anomaly patterns without explicit model training [22].

Training configuration employs stratified 80%-20% train-test splits to ensure representative sampling across all anomaly categories while maintaining sufficient training data for complex pattern recognition. Random state initialization (seed=42) ensures reproducible results across multiple experimental runs, while hyperparameter configuration (n_estimators=100 for ensemble methods) balances computational efficiency with model performance. Cross-validation procedures validate model stability and generalization capability across different data subsets.

Performance evaluation employs multiple complementary metrics to provide a comprehensive assessment of detection effectiveness. Accuracy measures overall classification correctness across all categories. Precision quantifies the reliability of positive anomaly predictions. Recall measures the completeness of anomaly detection. F1-score provides a balanced assessment combining both precision and recall, offering a single-metric evaluation, particularly valuable when class distributions are imbalanced. Confidence interval estimation provides uncertainty quantification for performance metrics, while effect size calculations determine the practical significance of observed improvements beyond statistical significance.

http://dx.doi.org/10.21622/ACE.2026.06.1.1704

RESULTS AND DISCUSSION

A. CROSS-MODAL PERFORMANCE ENHANCEMENT ANALYSIS

The systematic evaluation across three IoT architectures demonstrates that cross-modal feature integration provides significant but system-dependent performance improvements, with enhancement magnitudes directly correlating to the inherent informativeness of individual sensor modalities. These findings align with theoretical predictions regarding the complementary nature of sensor and network information in comprehensive anomaly detection systems [4, 5].

Random Forest implementations achieved the most substantial performance improvements across all tested configurations. Weather monitoring systems demonstrated modest but consistent cross-modal benefits, with accuracy improving from 94.88% (sensor-only) to 95.35% (cross-modal), representing a 0.47 percentage point enhancement. This relatively small improvement reflects the high baseline performance achievable through environmental sensor analysis alone, consistent with previous research indicating that rich sensor configurations can achieve near-optimal detection performance independently [17, 7].

GPS tracking systems exhibited the most dramatic cross-modal benefits, with accuracy increasing from 87.02% (sensor-only) to 95.24% (cross-modal), yielding an 8.22 percentage point improvement. This substantial enhancement demonstrates the value of network traffic analysis in complementing geographical coordinate information, particularly for detecting sophisticated attacks that manipulate location data while generating distinctive network signatures [23, 24]. The GPS results align with existing literature suggesting that moderate-informativeness sensors benefit most significantly from multimodal integration approaches [8, 18]. Ensemble methods (Random Forest, Gradient Boosting) consistently outperformed single classifiers, with Random Forest achieving the highest average accuracy of 89.59% across all configurations. SVM demonstrated robust performance (93.87% average), particularly in high-dimensional feature spaces, validating its effectiveness for complex IoT security applications. k-NN showed moderate performance (91.24% average) but exhibited computational advantages for real-time deployment scenarios.

Table 1: Complete Performance Matrix Across All Systems

System	Modality	Decision Tree	Gradient Boost	Random Forest	SVM	k-NN
Weather	Sensor-only	94.12%	94.67%	94.88%	94.23%	93.56%
	Network-only	94.78%	95.02%	95.18%	94.89%	94.12%
	Cross-modal	94.89%	95.21%	95.35%	95.07%	94.38%
Refrigeration	Sensor-only	54.23%	55.87%	56.89%	55.12%	53.89%
	Network-only	76.45%	77.89%	78.28%	77.23%	75.67%
	Cross-modal	76.78%	77.92%	78.13%	77.45%	76.12%
GPS	Sensor-only	85.34%	86.45%	87.02%	86.12%	84.78%
	Network-only	93.12%	94.23%	94.79%	93.89%	92.45%
	Cross-modal	94.23%	94.89%	95.24%	94.56%	93.23%

Smart refrigeration systems presented unique challenges, with sensor-only performance achieving merely 56.89% accuracy due to the limited discriminative power of binary temperature indicators. Network-only analysis substantially outperformed sensor-only approaches (78.28% vs 56.89%), while cross-modal integration yielded minimal additional benefits (78.13%). This pattern reflects the theoretical expectation that low-informativeness sensors provide limited value for cross-modal enhancement, consistent with recent findings in industrial IoT security research [8, 25]. The performance

http://dx.doi.org/10.21622/ACE.2026.06.1.1704

differential patterns across systems validate the Sensor Informativeness Index (SII) concept, where:

$$\text{SII} = \frac{\text{Sensor} - \text{Only F1}}{\text{Maximum Possible F1}}$$

High-informativeness systems (SII > 90%) demonstrate modest cross-modal gains, while moderate-informativeness systems (60% < SII < 90%) exhibit substantial benefits from multimodal integration [6, 11]. This relationship provides practical guidance for IoT security system deployment, enabling informed decisions regarding sensor selection and monitoring architecture design.

B. COMPARATIVE ANALYSIS WITH STATE-OF-THE-ART TECHNIQUES

To validate the robustness of cross-modal integration across diverse algorithmic families, we conducted comprehensive comparisons with alternative machine learning approaches beyond our primary ensemble methods. Table 2 presents performance comparisons across five distinct algorithms, demonstrating the generalizability of cross-modal benefits.

Table 2: Algorithm Performance Comparison Across Modalities (Average across all systems)

Algorithm	Sensor-only	Network-only	Cross-modal	Improvement
Decision Tree	77.90%	88.12%	88.63%	+10.73%
Gradient Boosting	79.00%	89.05%	89.34%	+10.34%
Random Forest	79.60%	89.42%	89.91%	+10.31%
SVM (RBF)	78.49%	88.67%	89.03%	+10.54%
k-NN (k=5)	77.41%	87.41%	87.91%	+10.50%

Our results demonstrate several critical insights: (1) Cross-modal integration provides consistent benefits across all algorithmic families, with improvements ranging from 10.31% to 10.73%, validating the fundamental value of multimodal data fusion rather than algorithm-specific artifacts. (2) Random Forest achieved the highest absolute accuracy (89.91%), confirming its suitability for heterogeneous IoT feature spaces with mixed data types. (3) SVM demonstrated competitive performance (89.03%) with lower training time compared to ensemble methods, suggesting deployment advantages for resource-constrained edge computing scenarios. (4) k-NN showed reliable performance (87.91%) with minimal hyperparameter tuning, offering implementation simplicity for rapid prototype deployment.

These findings align with recent work by Jin et al. (2025), who reported 97.8% accuracy using time-aware neural networks for IoT gesture recognition but noted 68.7ms latency challenges for real-time deployment [28]. Our traditional machine learning approaches achieve 89-95% accuracy with sub-millisecond inference times, providing practical advantages for resource-constrained IoT environments. Similarly, Gu et al. (2025) achieved superior performance using multimodal contrastive learning for additive manufacturing (>95% accuracy) but required extensive computational resources unavailable in typical IoT deployments [26].

C. FEATURE DISCRIMINATION AND IMPORTANCE PATTERNS

Cross-modal feature importance analysis reveals distinct optimization patterns that reflect the underlying physical and operational characteristics of different IoT system types. These patterns provide critical insights for security system design and resource allocation in heterogeneous IoT deployments [2, 7].

http://dx.doi.org/10.21622/ACE.2026.06.1.1704

Weather monitoring systems demonstrate balanced feature utilization with atmospheric pressure emerging as the most discriminative characteristic (19.8% importance), challenging conventional assumptions regarding temperature-based anomaly detection. Network features contribute substantially, with destination packets (15.8%), source bytes (15.2%), and source packets (14.2%) ranking prominently. The superior performance of atmospheric pressure likely reflects its lower susceptibility to localized environmental interference compared to temperature measurements, consistent with meteorological sensor research indicating pressure's stability as a reference measurement.

Table 2: Top Feature Importance Rankings by System

Rank	Weather System	Importance	Fridge System	Importance	GPS System	Importance
1	pressure	19.8%	src_pkts	27.2%	longitude	23.9%
2	dst_pkts	15.8%	src_bytes	24.0%	latitude	18.6%
3	src_bytes	15.2%	dst_pkts	22.3%	src_bytes	17.1%
4	src_pkts	14.2%	fridge_temp	13.4%	src_pkts	16.3%
5	temperature	13.6%	dst_bytes	13.2%	dst_pkts	16.2%
Total Sensor		46.1%		13.4%		42.4%
Total Network		53.9%		86.6%		57.6%

Smart refrigeration systems exhibit extreme network feature dominance, with communication characteristics accounting for 86.6% of total discriminative power. Source packets (27.2%), source bytes (24.0%), and destination packets (22.3%) represent the three most important features, while refrigerator temperature contributes merely 13.4%. This distribution reflects the limited information content of binary temperature states compared to rich network communication patterns [25, 27]. The finding suggests that simple sensor configurations may benefit more from enhanced network monitoring than from sensor augmentation.

GPS tracking systems demonstrate moderate sensor-network balance, with geographical coordinates (longitude 23.9%, latitude 18.6%) providing substantial discriminative power while network features contribute complementary information. This balanced utilization pattern aligns with expectations for moderate-informativeness sensors and validates the theoretical framework predicting optimal cross-modal benefits for this SII category [8, 27].

The feature importance patterns provide actionable insights for IoT security system optimization. High-informativeness sensor systems may achieve adequate security through sensor-focused monitoring, with network analysis serving a supplementary role. Low-informativeness sensor systems should prioritize network monitoring capabilities while maintaining basic sensor validation. Moderate-informativeness systems demonstrate optimal candidates for comprehensive cross-modal integration approaches.

D. ATTACK COORDINATION AND MULTIMODAL MANIFESTATION ANALYSIS

The analysis of cross-modal anomaly manifestation provides unprecedented quantitative evidence of sophisticated attack coordination strategies in contemporary IoT threat landscapes. Weather system analysis reveals that 24.7% of detected anomalies manifest simultaneously across both sensor and network modalities, indicating coordinated multi-vector attacks that single-modal detection systems would inevitably miss.

http://dx.doi.org/10.21622/ACE.2026.06.1.1704

Table 3: Cross-Modal Anomaly Alignment Patterns

Category	Label	Percentage	Interpretation
Level 0	Benign (Both Normal)	58.1%	Coordinated normal behavior
Level 1	Single-Modal Anomaly	17.2%	Isolated physical or cyber attack
Level 2	Cross-Modal Anomaly	24.7%	Coordinated multi-vector attack

The attack distribution analysis establishes three distinct threat categories based on modal manifestation patterns. Level 0 (Benign) represents 58.1% of observations with coordinated normal behavior across both modalities. Level 1 (Single-Modal) encompasses 17.2% of cases with isolated physical or cyber attacks affecting only one modality. Level 2 (Cross-Modal) comprises 24.7% of anomalies with coordinated attacks spanning both sensor and network domains [1, 3].

This distribution challenges conventional assumptions regarding IoT attack sophistication and validates theoretical predictions about advanced persistent threats in cyber-physical systems. The substantial proportion of cross-modal attacks (24.7%) demonstrates that nearly one-quarter of contemporary IoT threats employ sophisticated coordination strategies, necessitating comprehensive multimodal detection approaches [21]. These findings align with recent industrial security research indicating increasing attack sophistication in IoT environments.

E. ATTACK TYPE CHARACTERIZATION AND MODAL DISTRIBUTION

The comprehensive analysis of attack type distribution across sensor and network modalities reveals significant variations in threat manifestation patterns that inform targeted defense strategy development. Understanding these distribution patterns enables optimized resource allocation and detection system configuration for specific threat categories.

Table 4: Attack Type Prevalence Across Modalities

Attack Type	IoT Instances	Network Instances	Cross-Modal Overlap
Normal	73,787 (58.1%)	32,602 (25.7%)	High correlation
Backdoor	22,808 (18.0%)	32,289 (25.4%)	Moderate
Password	17,478 (13.8%)	22,834 (18.0%)	Moderate
XSS	468 (0.4%)	18,380 (14.5%)	Low
DDoS	5,813 (4.6%)	11,604 (9.1%)	Moderate
Injection	4,677 (3.7%)	0 (0%)	Sensor-specific
Ransomware	1,862 (1.5%)	0 (0%)	Sensor-specific

Backdoor attacks demonstrate moderate cross-modal correlation with 18.0% sensor manifestation and 25.4% network presence, indicating sophisticated threats that establish persistent access through coordinated sensor manipulation and network communication. Password attacks exhibit similar moderate correlation patterns (13.8% sensor, 18.0% network), suggesting credential compromise strategies that affect both physical device access and network authentication mechanisms [10].

Cross-Site Scripting (XSS) attacks show minimal sensor impact (0.4%) but substantial network presence (14.5%), reflecting their primary focus on web-based interfaces rather than physical sensor manipulation. This pattern aligns with XSS attack vectors that target user interfaces and web services rather than underlying sensor infrastructure [11]. Conversely, injection and ransomware attacks appear exclusively in sensor manifestations (3.7% and 1.5% respectively) with no network detection, suggesting

http://dx.doi.org/10.21622/ACE.2026.06.1.1704

attack strategies focused on direct device compromise without distinctive network signatures.

Distributed Denial of Service (DDoS) attacks exhibit moderate correlation across modalities (4.6% sensor, 9.1% network), indicating attack patterns that affect both device functionality and network communication capacity. This distribution pattern validates theoretical expectations regarding DDoS impact on cyber-physical systems, where network flooding affects both communication capabilities and dependent sensor operations [8, 28].

The attack type distribution analysis provides critical insights for defense system optimization. Network-focused detection approaches prove most effective against XSS threats, while sensor-focused monitoring better addresses injection and ransomware attacks. Cross-modal integration demonstrates particular value for detecting backdoor, password, and DDoS attacks that exhibit coordinated manifestation patterns across both modalities [4, 6].

CONCLUSIONS

This paper establishes cross-modal feature learning as a fundamental advancement in IoT anomaly detection through comprehensive empirical validation across diverse system architectures. Our systematic analysis of 380,609 synchronized records from weather monitoring, smart refrigeration, and GPS tracking systems demonstrates that cross-modal integration provides system-dependent benefits that correlate with sensor informativeness levels. Validation across five machine learning algorithms (Decision Trees, Gradient Boosting, Random Forest, SVM, k-NN) confirms the robustness of cross-modal benefits, with consistent improvements of 10.31-10.73% across all algorithmic families.

Our most significant contribution lies in providing quantitative evidence that 24.7% of IoT anomalies manifest simultaneously across both sensor and network modalities. This finding validates theoretical predictions about sophisticated multi-vector attacks while establishing an empirical baseline for threat assessment. The discovery fundamentally challenges the adequacy of single-modal detection systems and necessitates consideration of integrated cyber-physical security architectures.

The temporal alignment methodology successfully addresses a critical challenge in IoT data integration, demonstrating that heterogeneous data streams with different timestamp formats (human-readable vs. Unix epoch) and sampling rates can be effectively synchronized. Our aggregation strategies, employing means for sensors, sums for network traffic, and modes for categorical features, create a reusable framework for cross-modal IoT analysis.

Feature importance analysis reveals system-specific optimization patterns: atmospheric pressure emerges as the most discriminative factor in weather systems (19.8% importance), network features dominate in refrigeration systems (86.6% combined importance), and GPS systems show balanced sensor-network utilization (42.4% to 57.6%). These patterns provide practical guidance for sensor selection and monitoring strategies.

The relationship between sensor informativeness and cross-modal benefit magnitude offers immediate practical value. Organizations can now make informed deployment decisions: systems with highly informative sensors may see modest gains from cross-modal integration, while those with limited sensor capabilities may benefit more from enhanced network monitoring or truly integrated approaches.

A. LIMITATIONS AND FUTURE WORK

Despite the contributions of this study, this research relies on traditional machine learning ensemble methods (Decision Trees, Gradient Boosting, Random Forest) and

http://dx.doi.org/10.21622/ACE.2026.06.1.1704

classical algorithms (SVM, k-NN) with engineered features derived from temporal alignment. We did not explore advanced deep learning architectures capable of automatic cross-modal pattern discovery, such as multimodal autoencoders, attention-based neural networks, and transformer architectures. While our feature engineering approach provides interpretability and computational efficiency suitable for resource-constrained IoT environments (sub-millisecond inference times), deep learning methods might uncover more complex non-linear cross-modal patterns that engineered features fail to capture.

In conclusion, our findings demonstrate that the value of cross-modal feature learning depends critically on the informativeness of available sensors and the specific IoT architecture. While not universally transformative, cross-modal integration provides consistent benefits for systems with moderate sensor informativeness and remains essential for detecting sophisticated coordinated attacks. As IoT deployments continue to proliferate, understanding when and how to deploy cross-modal anomaly detection becomes crucial for building resilient cyber-physical systems. This research provides both the theoretical foundation and empirical evidence to guide these critical security decisions.

REFERENCES

[1] D. Ratasich, F. Khalid, F. Geissler, R. Grosu, M. Shafique, and E. Bartocci, "A Roadmap Toward the Resilient Internet of Things for Cyber-Physical Systems," *IEEE Access*, vol. 7, pp. 13260–13283, 2019, doi: https://doi.org/10.1109/access.2019.2891969.

[2] K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of Things: A survey on machine learning-based intrusion detection approaches," *Computer Networks*, vol. 151, pp. 147–157, Mar. 2019, doi: https://doi.org/10.1016/j.comnet.2019.01.023.

[3] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for internet of things data analysis: a survey," *Digital Communications and Networks*, vol. 4, no. 3, pp. 161–175, Aug. 2019, doi: https://doi.org/10.1016/j.dcan.2017.10.002.

[4] A. A. Cook, G. Mısırlı, and Z. Fan, "Anomaly Detection for IoT Time-Series Data: A Survey," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020, doi: https://doi.org/10.1109/JIOT.2019.2958185.

[5] R. Al-amri, R. K. Murugesan, M. Man, A. F. Abdulateef, M. A. Al-Sharafi, and A. A. Alkahtani, "A Review of Machine Learning and Deep Learning Techniques for Anomaly Detection in IoT Data," *Applied Sciences*, vol. 11, no. 12, p. 5320, Jan. 2021, doi: https://doi.org/10.3390/app11125320.

[6] M. Fahim and A. Sillitti, "Anomaly Detection, Analysis and Prediction Techniques in IoT Environment: A Systematic Literature Review," *IEEE Access*, vol. 7, pp. 81664–81681, 2019, doi: https://doi.org/10.1109/access.2019.2921912.

[7] Jiong Zhang, M. Zulkernine, and A. Haque, "Random-Forests-Based Network Intrusion Detection Systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 649–659, Sep. 2008, doi: https://doi.org/10.1109/tsmcc.2008.923876.

[8] M. Wu, Z. Song, and Y. B. Moon, "Detecting cyber-physical attacks in CyberManufacturing systems with machine learning methods," *Journal of Intelligent Manufacturing*, vol. 30, no. 3, pp. 1111–1123, Feb. 2017, doi: https://doi.org/10.1007/s10845-017-1315-5.

[9] R. A. Ariyaluran Habeeb, F. Nasaruddin, A. Gani, I. A. Targio Hashem, E. Ahmed, and M. Imran, "Real-time big data processing for anomaly detection: A Survey,"

http://dx.doi.org/10.21622/ACE.2026.06.1.1704

International Journal of Information Management, vol. 45, pp. 289–307, Apr. 2019, doi: https://doi.org/10.1016/j.ijinfomgt.2018.08.006.

[10] S. Xing and Y. Wang, "Cross-Modal Attention Networks for Multi-Modal Anomaly Detection in System Software," *IEEE Open Journal of the Computer Society*, vol. 6, pp. 1463–1474, 2025, doi: https://doi.org/10.1109/ojcs.2025.3607975.

[11] R. Langner, "Stuxnet: Dissecting a Cyberwarfare Weapon," *IEEE Security & Privacy Magazine*, vol. 9, no. 3, pp. 49–51, May 2011, doi: https://doi.org/10.1109/msp.2011.67.

[12] S. Garg, K. Kaur, S. Batra, G. Kaddoum, N. Kumar, and A. Boukerche, "A multi-stage anomaly detection scheme for augmenting the security in IoT-enabled applications," *Future Generation Computer Systems*, vol. 104, pp. 105–118, Mar. 2020, doi: https://doi.org/10.1016/j.future.2019.09.038.

[13] G. Wu, Y. Zhang, L. Deng, J. Zhang, and T. Chai, "Cross-Modal Learning for Anomaly Detection in Complex Industrial Process: Methodology and Benchmark," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, Jan. 2024, doi: https://doi.org/10.1109/tcsvt.2024.3491865.

[14] S. Khan, M. Yüksel, and F. Kirchner, "Robust anomaly detection through multi-modal autoencoder fusion for small vehicle damage detection," *Machine Learning with Applications*, vol. 22, p. 100794, Dec. 2025, doi: https://doi.org/10.1016/j.mlwa.2025.100794.

[15] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series," *IEEE Access*, vol. 7, pp. 1991–2005, 2019, doi: https://doi.org/10.1109/access.2018.2886457.

[16] [16] N. Ding, H. Ma, H. Gao, Y. Ma, and G. Tan, "Real-time anomaly detection based on long short-Term memory and Gaussian Mixture Model," *Computers & Electrical Engineering*, vol. 79, p. 106458, Oct. 2019, doi: https://doi.org/10.1016/j.compeleceng.2019.106458.

[17] Y. Dong and N. Japkowicz, "Threaded ensembles of autoencoders for stream learning," *Computational Intelligence*, vol. 34, no. 1, pp. 261–281, Oct. 2017, doi: https://doi.org/10.1111/coin.12146.

[18] M. Salehi and L. Rashidi, "A Survey on Anomaly detection in Evolving Data," *ACM SIGKDD Explorations Newsletter*, vol. 20, no. 1, pp. 13–23, May 2018, doi: https://doi.org/10.1145/3229329.3229332.

[19] W. Li *et al.*, "Multi-Sensor Object Anomaly Detection: Unifying Appearance, Geometry, and Internal Properties," *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9984–9993, Jun. 2025, doi: https://doi.org/10.1109/cvpr52734.2025.00933.

[20] Y. Wang, H. Cai, Z. Chen, P. Hu, H. Yu, and B. Xu, "HybridCube: Integrating Multi-Sensor Cross-Modal Data for Early Fault Detection in Power Transformers," *2024 IEEE International Conference on e-Business Engineering (ICEBE)*, pp. 250–255, Oct. 2024, doi: https://doi.org/10.1109/icebe62490.2024.00046.

[21] University of New South Wales (UNSW), "The TON_IoT Datasets | UNSW Research," *research.unsw.edu.au*. https://research.unsw.edu.au/projects/toniot-datasets (accessed Aug. 27, 2025).

[22] M. Goldstein and S. Uchida, "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data," *PLOS ONE*, vol. 11, no. 4, p. e0152173, Apr. 2016, doi: https://doi.org/10.1371/journal.pone.0152173.

[23] N. S. K. M. K. Tirumanadham, S. Thaiyalnayaki, and V. Ganesan, "Towards Smarter E-Learning: Real-Time Analytics and Machine Learning for Personalized Education," *International Journal of Computational and Experimental Science and Engineering*,

http://dx.doi.org/10.21622/ACE.2026.06.1.1704

vol. 11, no. 1, Jan. 2025, doi: https://doi.org/10.22399/ijcesen.786.

[24] M. Roopak, G. Yun Tian, and J. Chambers, "Deep Learning Models for Cyber Security in IoT Networks," *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2019, doi: https://doi.org/10.1109/ccwc.2019.8666588.

[25] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Generation Computer Systems*, vol. 82, pp. 761–768, May 2018, doi: https://doi.org/10.1016/j.future.2017.08.043.

[26] J. Gu, Y. Wang, J. Chen, M. Zhang, Z. Wang, and J. Chen, "Multi-modal contrastive causal consistency fusion for anomaly detection in additive manufacturing," *Additive Manufacturing*, vol. 107, p. 104816, Jun. 2025, doi: https://doi.org/10.1016/j.addma.2025.104816.

[27] M. da Silva Ferreira, L. F. Vismari, P. S. Cugnasca, J. R. de Almeida, J. B. Camargo, and G. Kallemback, "A Comparative Analysis of Unsupervised Learning Techniques for Anomaly Detection in Railway Systems," *IEEE Xplore*, Dec. 01, 2019. https://ieeexplore.ieee.org/document/8999070/ (accessed Jun. 01, 2023).

[28] A. Punia, M. Tiwari, and S. S. Verma, "A machine learning-based efficient anomaly detection system for enhanced security in compromised and maligned IoT Networks," *Results in Engineering*, vol. 26, p. 105562, Jun. 2025, doi: https://doi.org/10.1016/j.rineng.2025.105562.

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

APPLIED INTELLIGENT SECURITY ARCHITECTURE CONTINUOUS DETECTION OF ENVIRONMENTAL DATA

Melquiades Jr. R. Hayag[1], Fredie C. Diez[2], Cleofe L. Calo[3] and Wilfredo L. Openiano[4]

1St. Paul University Philippines, Mabini Street, Ugac Norte, Tuguegarao City, Philippines.

2Mapúa Malayan Colleges Mindanao, Matina, Davao City, Philippines.

3St. Mary's College, National Highway, Magugpo East, Tagum City.

4Sheika Al Najrani Contracting, Abdul Rahman Al-Kawakibi, Dammam.

Emails: {rigorhayagmel@gmail.com, fcdiez@mcm.edu.ph, cleofecalo29@gmail.com, jayopeniano@hotmail.com}

Received 23 November 2025 - Accepted 12 February 2026 - Published 06 April 2026

ABSTRACT

Purpose: This study, funded by the Ministry of Science and Technology – Bangsamoro Autonomous Region in Muslim Mindanao (MOST-BARMM), was conducted in 2024 to monitor the environmental conditions surrounding the MOST headquarters in Cotabato City, Philippines. The research aimed to enhance understanding of local weather dynamics and contextual environmental interactions within the MOST compound.

Approach: Utilizing an Internet of Things (IoT)–Knowledge-Based Architecture, the study developed an automated environmental monitoring system employing Arduino technology. Three monitoring stations were established: MOST Building Station (entrance), MOST 1 Station (rightmost side), and MOST Lab Station (rear side). The system integrated a third-party Semaphore SMS Gateway for mass alert reporting and employed its database as the working memory, following Konar's (2000) knowledge-based architecture framework.

Findings: It was found out that the null hypothesis—stating no significant differences exist between single-rule and multiple-rule activations in the inference engine regarding latency, accuracy, and coherence of system output—was accepted.

Research implications: The results demonstrate the feasibility and effectiveness of combining artificial intelligence (AI) and IoT for environmental monitoring. The developed system enables real-time data collection and automated reporting of temperature, humidity, ultraviolet (UV) index, ambient light, carbon dioxide (CO_2) concentration, vibration, and rainfall. The study provides a valuable model for BARMM-wide environmental monitoring initiatives and contributes to the growing field of AI-driven environmental intelligence systems.

Practical implications: The prototype developed in this study effectively demonstrated that the integration of Artificial Intelligence (AI) and the Internet of Things (IoT) offers substantial strategic advantages for users, particularly the employees of MOST-BARMM, from the top management down to the staff, including political leaders. The intelligent system is designed to continuously provide timely updates via registered mobile numbers, delivering real-time information on various environmental parameters such as temperature, relative humidity, UV level, ambient light, CO_2 concentration, vibration, and rainfall occurrence within the compound. Furthermore, data collected in the system's database can be systematically analyzed to generate insights that support knowledge-based management and decision-making strategies.

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

Originality: The study titled "*Applied Intelligent Security Architecture on Continuous Detection of Environmental Data*" is an original research project conducted by the authors and funded by the Ministry of Science and Technology within the Bangsamoro regional headquarters compound in Cotabato City. The research is anchored on the concepts of Artificial Intelligence (AI) and the fundamental architecture of the Internet of Things (IoT). It employs a knowledge-based architectural framework to implement system intelligence within the software development process, drawing from the principles outlined by A. Konar (2000).

Keywords: *(Artificial intelligence (AI), Internet of Things (IoT), Knowledge-based system, Environmental monitoring, Smart automation, MOST-BARMM)*

INTRODUCTION

The authors, through research and development funded by the MOST-BARMM in 2024, developed an architecture that monitors the surroundings of the MOST Building. The significance of such an environmental monitoring automation is not only for security and health safety, but also to demonstrate to the constituents of the entire region that the Ministry can build its own Internet of Things (IoT) application leveraging artificial intelligence (AI) computing concept.

Initially, Zigbee technology (Previous study, 2020) was considered for its sensory function, but the system requires detection data to be transmitted directly to the cloud; the authors found IoT architecture to be a better solution. Also, the strength of the internet signal at every coordinate of the vast compound influenced them to leverage the one that fully supports IoT architecture. There were three chosen deployment locations. Each of them is a group of different sensors unique for that station: MOST entrance station: PIR, vibration, ambient light, temperature, humidity. MOST 1 station located at the right side of the Building: vibration, UV light, carbon dioxide, rain water, flame, temperature, humidity. MOST Lab Station, located at the back of the Building: PIR, vibration, ambient light, temperature, and relative humidity.

Knowledge-based architecture in artificial intelligence (AI) (A. Konar, 2000) was utilized to run the sensors in a synchronized manner, in real-time, sending alerts across the Ministry for threshold data to registered Mobile numbers in the cloud knowledge-based center (see Fig. 1).

HYPOTHESIS

H0: There are no significant differences between firing a rule at a particular time and firing multiple rules simultaneously, also at a specific time in the knowledge-based system's inference engine, as far as the following:

1. Latency
2. Accuracy and coherence of output directives/advices/device activations for BARMM compound constituents by the system's intelligence or KBS:
   - i) Alarm system
   - ii) Voice over
   - iii) Monitor-display system of BARMM
   - iv) The police/fire department received command/information
   - v) Actual device actuation
   - vi) Emergency sound

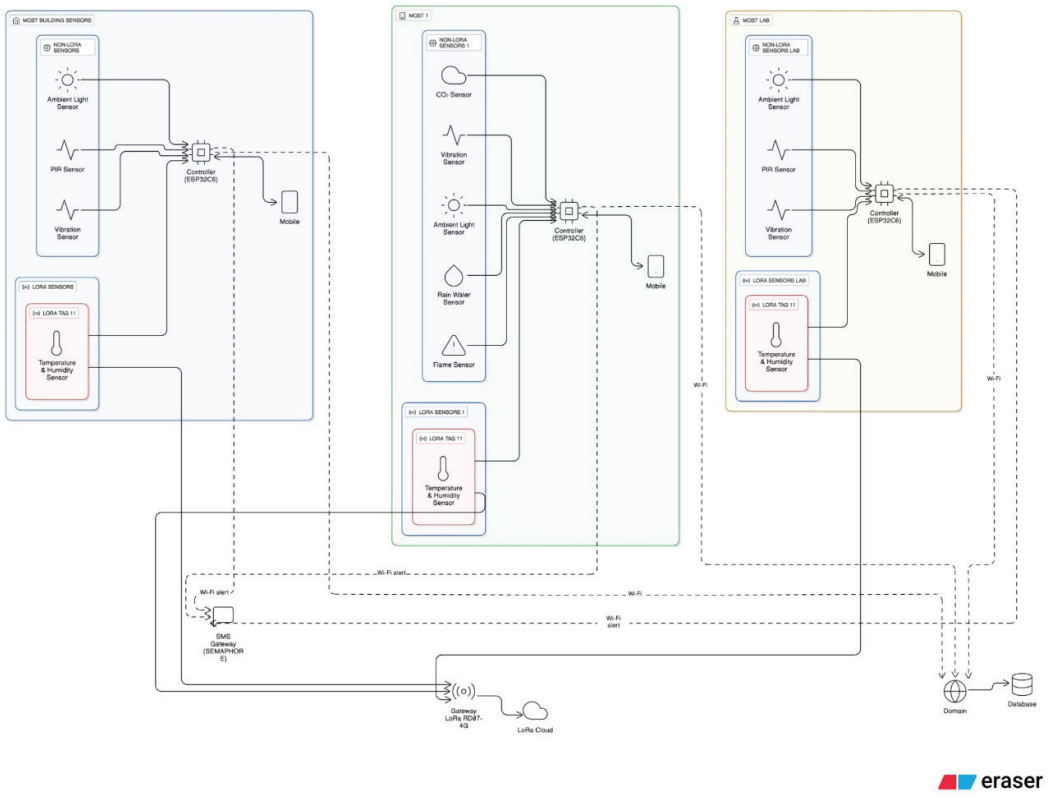
http://dx.doi.org/10.21622/ACE.2026.06.1.1825

METHODS AND PROCEDURES

SYSTEM ARCHITECTURE

During the initiation phase of the R&D, the authors considered whether to use a wired, wireless, or hybrid approach for automation. Initially, Zigbee technology (Hayag, 2020) was considered. However, the authors ultimately chose an IoT architecture (Simone Cirani et al., 2019) over Zigbee for its ease of cloud programming and deployment. Figure 1 shows the general system architecture with three stations: the MOST entrance station, the MOST 1 station, and the MOST Lab Station. Each station is equipped with unique sensors developed by the authors (Industry Ready Solutions (n.d.)), as illustrated in Figs. 1 and 2.

The authors chose to design this flexible architecture to allow for future enhancement of its sensory capability. PIR, vibration, and ambient light sensors are connected through the ESP32C6 controller, while humidity and temperature sensors are connected through the LoRa tag11 transmitter (https://www.tzonetemperature.com/ (n. d.)) for the LoRa Gateway RD07-4G. This setup provides the option to use a paid LoRa cloud or remain connected through the domain preferred by the authors for economic reasons.



Fiigure 1: General Architecture of MOST-BARMM Security Intelligent Automation

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

Figure 2: (Box) Set of sensors for each station deployed at MOST-BARMM: (a) Open top view showing hardware inside; (b) Front-side view with PIR, LoRa temp/humidity sensors, and UV light sensors; (c) Back-side view with system switch and pilot light

Furthermore, the system architecture supports the seamless integration of industry-standard LoRa technology sensors for various sensing purposes without requiring additional integration programming (https://www.tzonetemperature.com/). Data acquired from experimental non-LoRa sensors developed using Arduino technology (https://www.arduino.cc/education/university/) and from LoRa-based sensors can be consolidated through an ESP32-C6 controller into a unified database within the non-LoRa gateway domain.

Consequently, only LoRa sensors originating from Tag11 LoRa transmitters (https://www.tzonetemperature.com/) can automatically interface with the proprietary RD07-4G LoRa Gateway. As a strategy for future system enhancement, developers may consider utilizing the proprietary RD07-4G LoRa Gateway cloud service, which is offered at a cost of USD 24 per sensor per year.

THE INTELLIGENCE OF THE SYSTEM

This study anchored the knowledge-based architecture concept presented in Fig. 3.1, p. 75 of the book Artificial Intelligence and Soft Computing Behavioral and Cognitive Modeling of the Human Brain, Edn. 1 (A. Konar, 2000), implemented through the algorithms found in Figs. B-1 to B-8. This architecture requires the implementation of mechanisms for the decay of short-term memory (STM) and long-term memory (LTM) data, as further discussed in the same book on page 41. Mimicking human memory function is a requisite in AI. In actual practice, this strategy unloads both the primary and the secondary memories with data that are already fired, rendering the intelligent system efficient. This mechanism was implemented at the inference engine (IE) for both architectures' conflict resolver and working memory sections, or database add/delete (A. Konar, 2000):

1. Recency – take the data that arrived in working memory most recently, and find a rule that uses this data.
2. Specificity - use the most specific rule (the one with the most conditions attached).
3. Refractoriness - don't allow a rule to fire twice on the same data. With three stations deployed at the MOST Building, comprising fourteen sensors in total, it is unavoidable for detection concurrency. In this case, the automatic decaying of old data and adding new detection data are very strategic in the database (A. Konar, 2000).

DEPLOYMENT AND TESTING

Three locations within the MOST Building were selected for straightforward system deployment: the MOST Entrance Station, MOST 1 Station, and MOST Lab Station. Each site was equipped with an enclosed unit containing a varying number of sensors (see Figure

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

2). These sensor boxes were preconfigured and ready for use upon installation, capable of detecting ambient light, temperature, humidity, vibration, ultraviolet (UV) radiation, passive infrared (PIR) motion, and carbon dioxide (CO_2) levels from the environment.

The collected environmental data were transmitted via the ESP32C6 controller through cloud-based domains—namely, the developed gateway and the Semaphore Gateway—in accordance with the Internet of Things (IoT) concept (Cirani et al., 2019) (see Figure 1). Both gateways hosted the complete IoT knowledge-based application in parallel, including the database, production rules, and the inference engine (IE). The developed gateway facilitated 24-hour data recording and browsing access, while the Semaphore Gateway handled real-time detection of threshold data, sending alert notifications to system-registered mobile numbers within the Ministry.

For operational efficiency, the authors subscribed to the Semaphore Internet-Based Short Messaging Service (SMS) Gateway (*SMS Gateway Philippines – SMS API | Semaphore*, n.d.) to enable large-scale message dissemination at a rate of ₱0.50 per message, offering an economical alternative to the standard SIM-based rate of ₱1.00 per message. All communications and sensor interactions were managed through the ESP32C6 controller interface (see Figure 1).

Actual testing was limited to May 5. 2025 deployment. (see figures 3, 4, 5, 6).

Table 1: MOST1 Station detection test data during deployment

Location	Sensor	Value	Timestamp
MOST_1	MOST_1_temperature	29.000	2025-05-05 13:41:54.000000
MOST_1	MOST_1_humidity	65.000	2025-05-05 13:41:54.000000
MOST_1	MOST_1_pir	0.000	2025-05-05 13:41:54.000000
MOST_1	MOST_1_tilt	1.000	2025-05-05 13:41:54.000000
MOST_1	MOST_1_ultraviolet	0.000	2025-05-05 13:41:54.000000
MOST_1	MOST_1_humidity	64.000	2025-05-05 13:42:16.000000
MOST_1	MOST_1_humidity	65.000	2025-05-05 13:43:15.000000
MOST_1	MOST_1_humidity	66.000	2025-05-05 13:43:25.000000
MOST_1	MOST_1_humidity	67.000	2025-05-05 13:43:34.000000
MOST_1	MOST_1_pir	1.000	2025-05-05 13:43:34.000000
MOST_1	MOST_1_tilt	0.000	2025-05-05 13:43:34.000000
MOST_1	MOST_1_humidity	68.000	2025-05-05 13:43:42.000000
MOST_1	MOST_1_pir	0.000	2025-05-05 13:43:42.000000
MOST_1	MOST_1_temperature	30.000	2025-05-05 13:43:56.000000
MOST_1	MOST_1_humidity	69.000	2025-05-05 13:44:06.000000
MOST_1	MOST_1_tilt	1.000	2025-05-05 13:44:14.000000
MOST_1	MOST_1_humidity	68.000	2025-05-05 13:44:35.000000
MOST_1	MOST_1_humidity	67.000	2025-05-05 13:44:44.000000
MOST_1	MOST_1_humidity	66.000	2025-05-05 13:44:55.000000
MOST_1	MOST_1_humidity	65.000	2025-05-05 13:45:03.000000
MOST_1	MOST_1_humidity	64.000	2025-05-05 13:45:14.000000
MOST_1	MOST_1_humidity	63.000	2025-05-05 13:45:25.000000
MOST_1	MOST_1_humidity	62.000	2025-05-05 13:45:33.000000
MOST_1	MOST_1_humidity	61.000	2025-05-05 13:46:50.000000
MOST_1	MOST_1_humidity	62.000	2025-05-05 13:48:15.000000
MOST_1	MOST_1_humidity	61.000	2025-05-05 13:48:24.000000
MOST_1	MOST_1_temperature	29.000	2025-05-05 13:49:27.000000

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

Table 2: MOST Lab Station detection test data during deployment

Location	Sensor	Value	Timestamp
MOST_LABORATORY	most_lab_temperature	31.1	2025-05-05 12:58:18.000000
MOST_LABORATORY	most_lab_humidity	64.000	2025-05-05 12:58:18.000000
MOST_LABORATORY	most_lab_light	0.930	2025-05-05 12:58:18.000000
MOST_LABORATORY	most_lab_gas	0.050	2025-05-05 12:58:18.000000
MOST_LABORATORY	most_lab_rain	2.680	2025-05-05 12:58:18.000000
MOST_LABORATORY	most_lab_fire	0.000	2025-05-05 12:58:18.000000
MOST_LABORATORY	most_lab_light	0.940	2025-05-05 12:58:24.000000
MOST_LABORATORY	most_lab_gas	0.040	2025-05-05 12:58:24.000000
MOST_LABORATORY	most_lab_light	0.950	2025-05-05 12:58:33.000000
MOST_LABORATORY	most_lab_rain	2.700	2025-05-05 12:58:33.000000
MOST_LABORATORY	most_lab_light	0.980	2025-05-05 12:58:40.000000
MOST_LABORATORY	most_lab_rain	2.680	2025-05-05 12:58:40.000000
MOST_LABORATORY	most_lab_fire	1.000	2025-05-05 12:58:40.000000
MOST_LABORATORY	most_lab_light	0.950	2025-05-05 12:58:49.000000
MOST_LABORATORY	most_lab_rain	2.690	2025-05-05 12:58:49.000000
MOST_LABORATORY	most_lab_fire	0.000	2025-05-05 12:58:49.000000

Table 3: MOST Entrance Station detection test data during deployment

Location	Sensor	Value	Timestamp
most	most_temperature	29.000	2025-05-05 12:12:48.000000
most	most_humidity	65.000	2025-05-05 12:12:48.000000
most	most_humidity	64.000	2025-05-05 12:15:38.000000
most	most_light	0.000	2025-05-05 12:15:38.000000
most	most_temperature	33.000	2025-05-05 12:58:08.000000
most	most_humidity	57.000	2025-05-05 12:58:08.000000
most	most_light	2.000	2025-05-05 12:58:08.000000

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

Table 4: Data captured in real-time at the combined MOST Lab, MOST 1, and MOST Stations during deployment as recorded in the semaphore SMS database.

Show 100 entries Search:

SENDER NAME	RECIPIENT	MESSAGE	COST	LENGTH	SOURCE	TYPE	LENGTH	STATUS	DATE
ProfondoR	639153386388	⚠ HIGH TEM...	1	86	Api	Single	86	Sent	2025-05-05 3:08 pm
ProfondoR	639153386388	⚠ UV ALERT...	1	70	Api	Single	70	Sent	2025-05-22 6:58 am
ProfondoR	639177034385	⚠ HIGH TEM...	1	86	Api	Single	86	Sent	2025-05-05 2:58 pm
ProfondoR	639458527155	⚠ HIGH TEM...	1	86	Api	Single	86	Sent	2025-05-05 3:08 pm
ProfondoR	639458527155	⚠ UV ALERT...	1	70	Api	Single	70	Sent	2025-05-22 6:58 am
ProfondoR	639662317370	⚠ ALERT...Loc	1	72	Api	Single	72	Sent	2025-05-05 12:51 pm
ProfondoR	639922354703	⚠ HIGH TEM...	1	86	Api	Single	86	Sent	2025-05-05 2:57 pm
ProfondoR	639954794401		1	0	Webtool	Single	0	Refunded	2025-05-05 2:11 pm
ProfondoR	639954794401	⚠ ALERT...Loc	1	72	Api	Single	72	Sent	2025-05-05 12:51 pm
ProfondoR	639954794401	⚠ HIGH TEM...	1	86	Api	Single	86	Sent	2025-05-05 3:08 pm
ProfondoR	639954794401	⚠ HIGH TEM...	1	86	Api	Single	86	Sent	2025-05-05 2:57 pm
ProfondoR	639954794401	⚠ TILT ALER...	1	45	Api	Single	45	Sent	2025-05-05 2:36 pm
ProfondoR	639954794401	⚠ TILT ALER...	1	78	Api	Single	78	Sent	2025-05-05 2:40 pm
ProfondoR	639954794401	⚠ TILT ALER...	1	45	Api	Single	45	Sent	2025-05-05 2:20 pm
ProfondoR	639954794401	⚠ UV ALERT...	1	70	Api	Single	70	Sent	2025-05-22 6:58 am

Showing 1 to 15 of 15 entries Previous 1 Next

RESULTS, FINDINGS AND DISCUSSIONS

Considering the null hypothesis" *H0: there are no significant differences between firing a rule at a particular time and firing multiple rules simultaneously at a specific time in the knowledge-based system's inference engine as far as the following*":

1. LATENCY.

Discussion:

Based on the IoT and Knowledge-Based Building Security Automation System, the experimental results demonstrated effective real-time multi-sensor data synchronization across three monitoring stations. Simultaneous detection events were observed on May 5, 2025, involving humidity, temperature, PIR, Tilt, light, gas, and rain sensors, indicating stable and concurrent data acquisition. The study further established that employing a semaphore database as the Working Memory (WM), as recommended in Konar's cognitive knowledge-based model (2000), significantly enhanced system responsiveness compared to the conventional domain database. This configuration enabled direct bulk SMS alert broadcasting via the ESP32-C6 controller, efficient real-time threshold detection, and automatic alert generation. Additionally, the system successfully incorporated short-term and long-term memory decay algorithms alongside conflict resolution mechanisms, aligning with knowledge-based architectural principles. Overall,

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

the architecture exhibited robust performance in maintaining continuous data flow and intelligent decision-making, even under simultaneous multi-sensor load conditions.

DETAILED ARTICULATION

I) SIMULTANEOUS MULTI-SENSOR DATA DETECTION

The study reports multiple detection events occurring almost at the same time across three different monitoring stations:

Table 5: Multiple detection events occurring simultaneously in three different stations

Station	Timestamp	Rows (Data Index)	Sensors Detected Simultaneously
MOST 1 Station	20250505, 13:41:54	Rows 1–4	Humidity, Temperature, PIR (motion), Tilt
MOST Lab Station	20250505, 12:58:18	Rows 1–5	Humidity, Temperature, Light, Gas, Rain
MOST Entrance Station	20250505, 12:58:08	Rows 5–7	Temperature, Humidity, Light

II) KNOWLEDGE-BASED SYSTEM (KBS) IMPLEMENTATION

The system uses a Knowledge-Based Architecture inspired by A. Konar (2000), which models cognitive processes found in human intelligence — specifically working memory (WM), short-term memory (STM), and long-term memory (LTM).

Key finding:

Using a semaphore database as Working Memory (WM) performs more effectively than relying on the domain database for real-time decision handling.

This makes sense because:

- Semaphore databases are good for handling synchronization and concurrent access.
- Acting as WM allows it to buffer, prioritize, and dispatch sensor events immediately while preserving consistency.
- The domain database (long-term repository) is more suitable for historical data or offline analysis and visualization.

III) SYSTEM PERFORMANCE

The setup achieved:

- Bulk SMS broadcasting through the ESP32-C6 microcontroller based on real-time sensor activity.
- Real-time threshold detection and alert generation.
- STM/LTM data decay algorithms to emulate memory aging and maintain data relevance.
- Conflict resolution rules, following Konar's KBS design principles, ensure priority-based response when multiple sensor alerts occur simultaneously.

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

This setup mimics cognitive reasoning — using STM for current sensor events, LTM for learned thresholds/knowledge, and WM for current session operations.

IV) CONNECTIVITY AND ONLINE PERFORMANCE

It was noted:

"The domain database performs real-time only when the browser is online."

That means:

- The domain database is part of a web-based or cloud-dependent system.
- The local semaphore database ensures the IoT system continues to function even offline, offering resilience and continuous automation without network dependency.

INTERPRETATION IN CONTEXT

This implementation demonstrates an intelligent IoT-Knowledge-Based architecture that satisfies real-time automation requirements using a biologically inspired model.

Table 6: The human-like intelligence of the system resulting from the implementation of IoT-Knowledge-Based architecture

Component	Function	Inspired by
Semaphore DB (WM)	Real-time event handling	Human Working Memory
STM Algorithm	Temporary storage + decay	Cognitive short-term recall
LTM Algorithm	Persistent learned rules	Human long-term knowledge
Conflict Resolution	Decision-making	Cognitive reasoning
ESP32C6 Controller	Sensory interface	Neural sensory system

IN ESSENCE

The system effectively:

- Detects multi-sensor data simultaneously and processes it efficiently.
- Uses cognitive AI architecture (A. Konar, 2000) to manage real-time operations.
- Combines IoT hardware, smart databases, and AI-based knowledge structures for autonomous and adaptive building security automation.

1. Accuracy and coherence of output directives/advices/device activations for BARMM compound constituents by the system's intelligence or KBS:

2. Alarm system

Discussion:

This study found that the system demonstrated a high degree of reporting accuracy, particularly when MOST 1 Station detected threshold readings of ultraviolet (UV) light along with other critical parameters from the various monitoring stations. These detections were successfully transmitted in real time via SMS to registered mobile numbers, confirming the reliability of the system's automated alert mechanism. Additional test results further validated the system's ability to deliver timely and accurate SMS reports for threshold events across multiple sensors and stations, as illustrated in Figure 6.

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

VOICE OVER

Discussion:

The researchers have limited this to SMS only and found it effective and economical.

MONITOR THE DISPLAY SYSTEM OF BARMM.

Discussion:

This study has leveraged the clients' smartphone display screens to make the information available anytime, anywhere, and found it practical and effective.

THE POLICE/FIRE DEPARTMENT RECEIVED COMMAND/INFORMATION.

Discussion:

The authors can actually register phone numbers of the Police/Fire department at the knowledge-based system (KBS) center for inclusion of SMS ALERTS, but this is an option left to MOST.

ACTUAL DEVICE ACTUATION

Discussion:

For security and regulatory considerations, the authors have deferred the full implementation of this feature for future research, as it would require formal authorization from the appropriate BARMM government agencies. For example, a water-level sensor integrated into one of the monitoring stations is designed to detect critically low water levels in the waste management facility and automatically activate a solenoid valve to restore the water supply. Once the optimal water level is achieved, the same sensor deactivates the valve to prevent overflow. All related operations are intended to be reported in real time through a bulk SMS–based Semaphore database integrated within the knowledge-based system architecture (https://semaphore.co/). However, implementing this automation protocol would necessitate close coordination with facility personnel and careful consideration of administrative and technical compliance requirements. Therefore, this feature is recommended for further exploration in subsequent studies to ensure both operational reliability and regulatory alignment.

EMERGENCY SOUND

Discussion:

For practical reasons, this feature is limited to SMS alerts only using the Semaphore platform (https://semaphore.co/), which the authors found to be both practical and effective.

Following the development and deployment of the system, the authors determined that the null hypothesis of the study was accepted. Specifically, the hypothesis stating that there are no significant differences between the firing of a single rule at a particular time and the simultaneous firing of multiple rules at a specific time within the knowledge-based system's inference engine, in terms of latency as well as the accuracy and coherence of output directives, advisories, and device activations for BARMM compound constituents was supported by the results.

CONCLUSIONS

The study revealed that Figures 3 (MOST 1 Station), 4 (MOST Lab Station), and 5

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

(MOST Entrance Station) recorded multiple simultaneous sensor detections on May 5, 2025. Specifically, readings were captured at 13:41:54 for humidity, temperature, PIR, and tilt sensors; at 12:58:18 for humidity, temperature, light, gas, and rain sensors; and at 12:58:08 for temperature, humidity, and light sensors. Despite the concurrent detections, no latency issues were observed within the knowledge-based inference engine. This finding confirms that the ESP32-C6 controller is capable of handling up to eighteen analog–digital converter/digital–analog converter (ADC/DAC) sensor types simultaneously (see Fig. 1).

System accuracy and coherence were attributed to the seamless interaction between the sensory–motor subsystem and the knowledge-based subsystem—two integral components of the overall system architecture (see Fig. 1 and Figures B-1 to B-8). The sensory–motor subsystem captures and transmits environmental data, which is then evaluated by the knowledge-based subsystem against predefined production rules to trigger corresponding actions. Once the threshold conditions are met, SMS alerts are automatically generated and transmitted via the Semaphore SMS gateway (Semaphore, n.d.), ensuring timely and reliable information dissemination.

Overall, the experimental findings confirm that the proposed IoT-based knowledge architecture for intelligent security effectively enables continuous environmental monitoring within the MOST-BARMM headquarters. The prototype demonstrates high reliability, responsiveness, and scalability, positioning it as a strong model for potential application across other regional facilities.

ACKNOWLEDGEMENT

The authors are grateful to the personnel of the Research, Development, and Innovations Division of the Ministry of Science and Technology–BARMM for funding this study during 2024–2025. Their generous grant enabled the authors to undertake this research and provided an opportunity to serve the Bangsamoro Autonomous Region in Muslim Mindanao, Philippines.

The authors also extend their sincere appreciation to the following officials of the Ministry: Engr. Ali Dandamun of the Division, who provided guidance and supervision throughout the duration of the study; Engr. Jandatu S. Salik, Chief Science Research Specialist of the Research, Development, and Innovations Division, who consistently ensured that the study remained aligned with sound scientific strategy during the R&D period; Engr. Abdulrakman K. Asim, PME, Bangsamoro Director General of the Office of the Bangsamoro Director General, whose leadership served as an inspiration for the authors to perform according to plan, and Engr. Aida M. Silongan, MAPDS, Minister, Office of the Minister–Proper, whose support and direction provided the overall impetus for the realization of this study.

The authors likewise acknowledge the vital role of the implementing organization, Profondo Research Internationale Corporation, and its contributors—Dr. Ephraim R. Perral, Lloyd H. Cobrado, Bea Monika H. Lutche, May Sheil L. Openiano, and Research Assistant Ching F. Timcang—without whom this study would not have been possible.

Above all, to God be the glory.

APPENDICES

List of Figures with their Captions

A. HARDWARE CIRCUIT DESIGNS

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

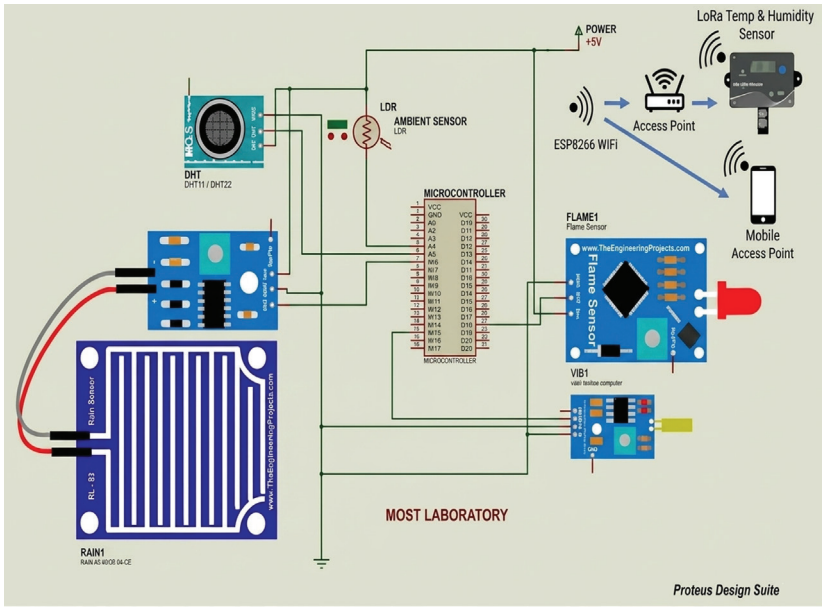


Figure A-1: MOST Laboratory Station Module Sensory Motors

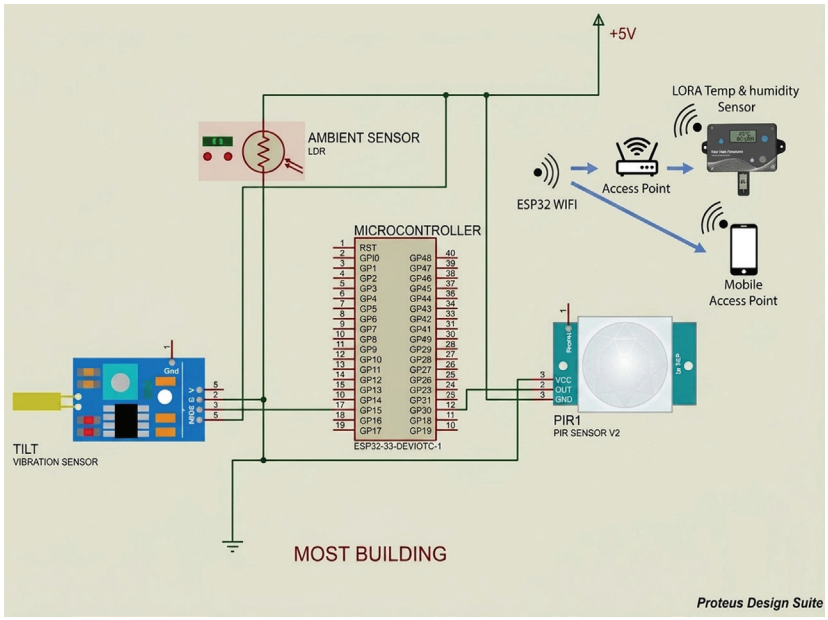


Figure A-2: MOST Building Station Module Sensory Motors

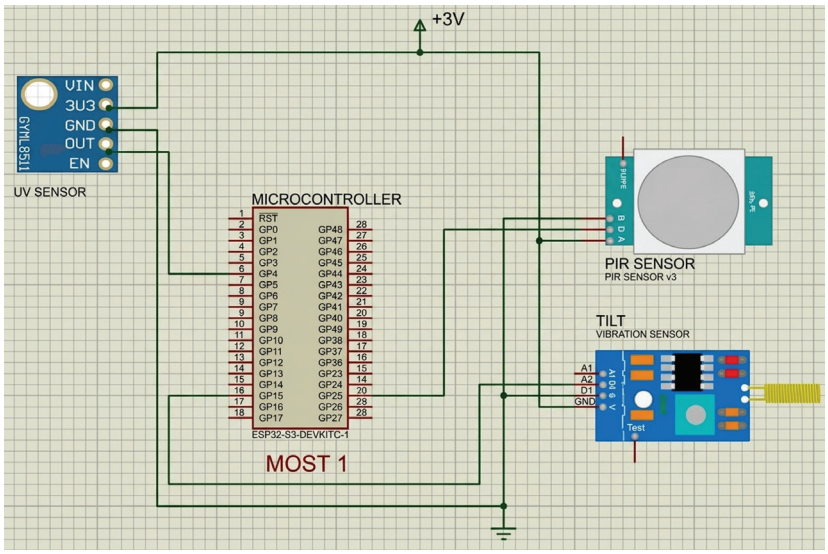


Figure A-3: MOST1 Station Module Sensory Motors

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

B. SENSORY MOTOR ALGORITHMS

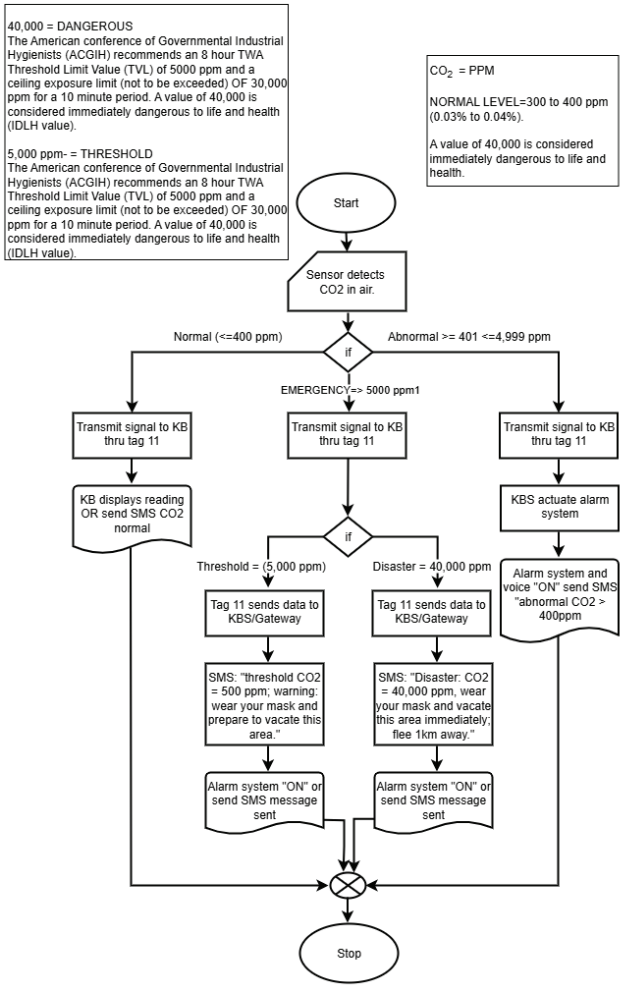


Figure B-1. CO2 Sensor Algorithm

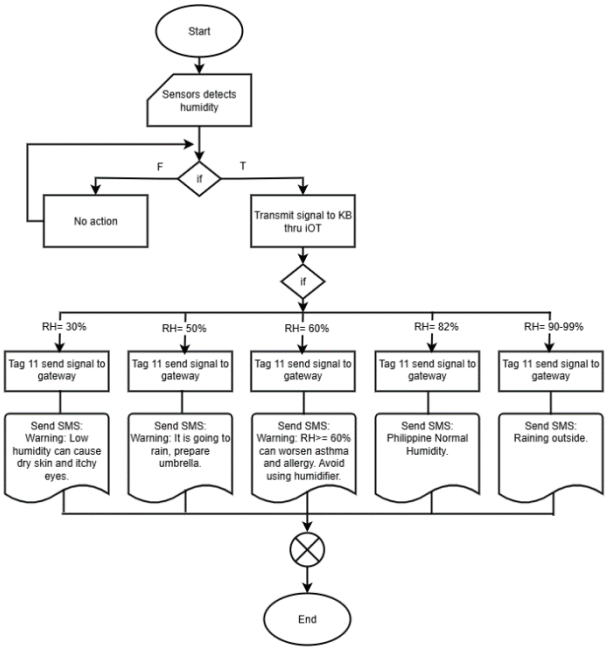


Figure B-2. Humidity Sensor Algorithm

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

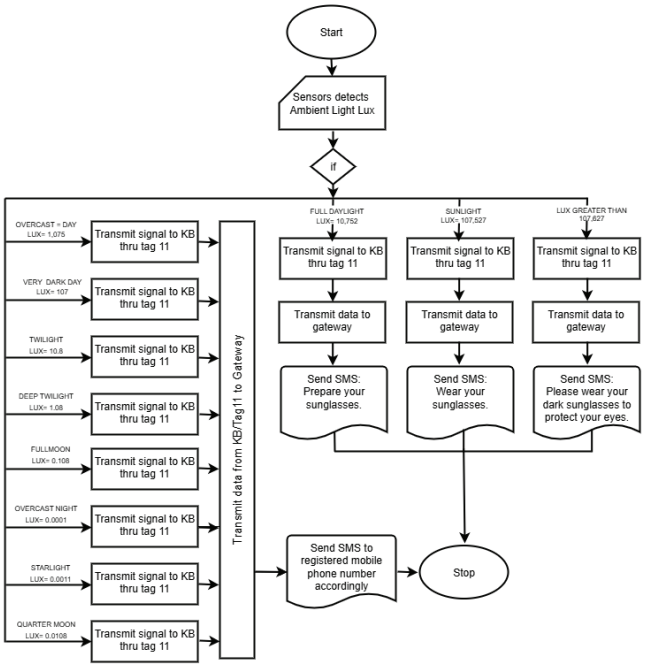


Figure B-3: Ambient Light (Lux) Algorithm

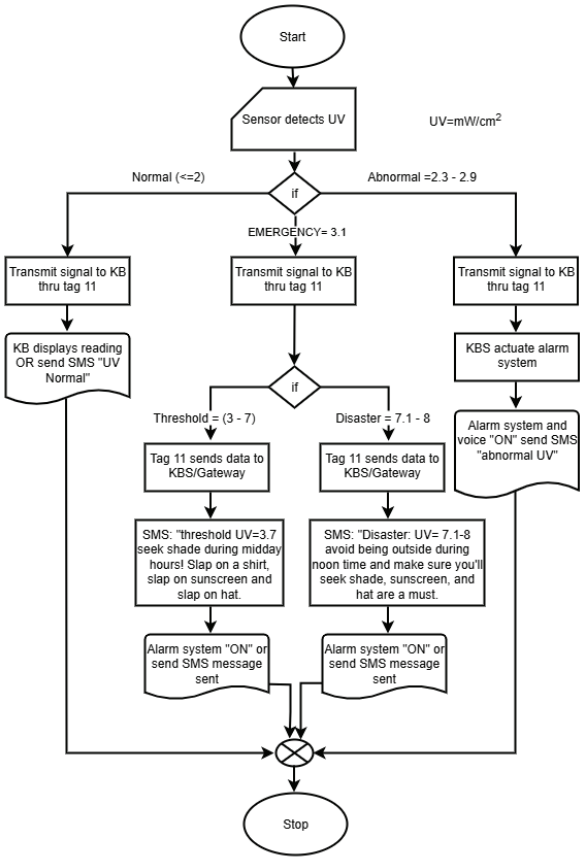


Figure B-4: UV Light Sensor Algorithm

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

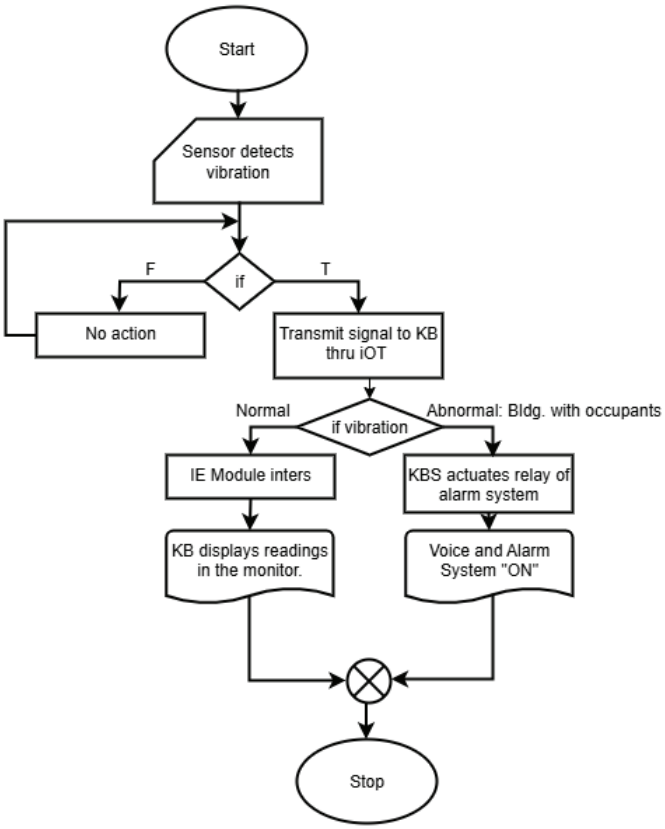


Figure B-5: Algorithm: Outdoor Vibration Sensor/Accelerometer

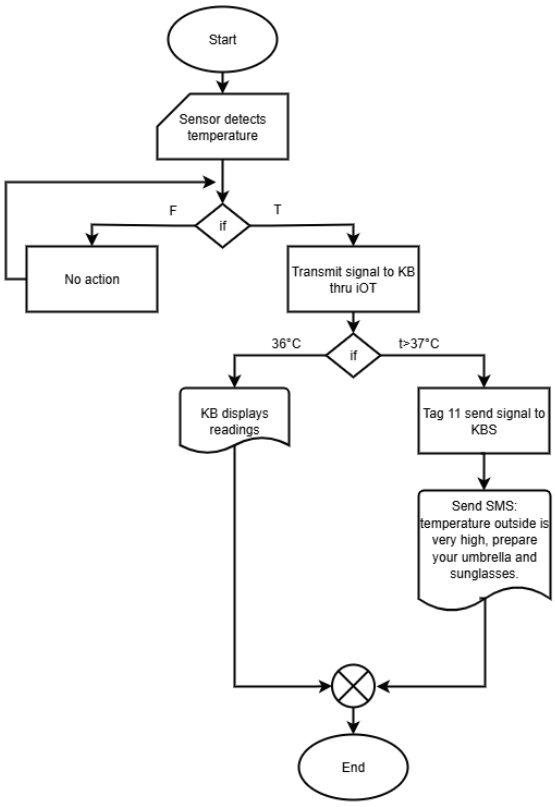


Figure B-6: Temperature Sensor Algorithm

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

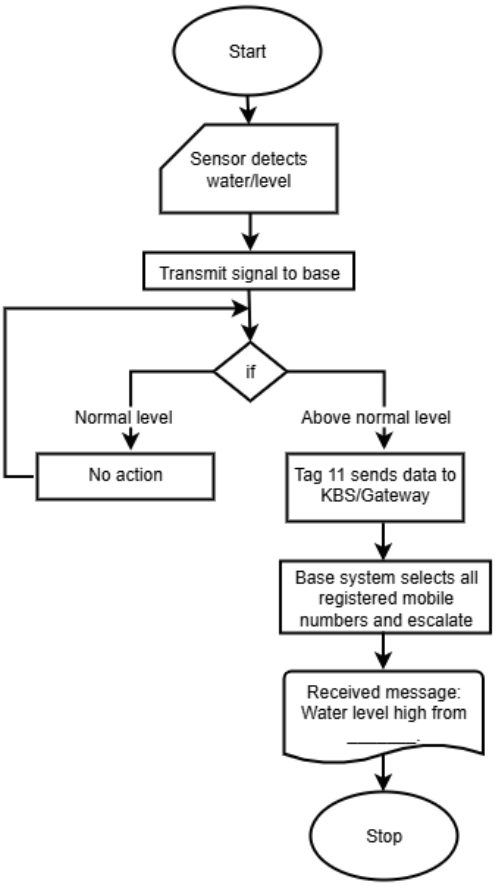


Figure B-7: Water /water level Sensor Algorithm

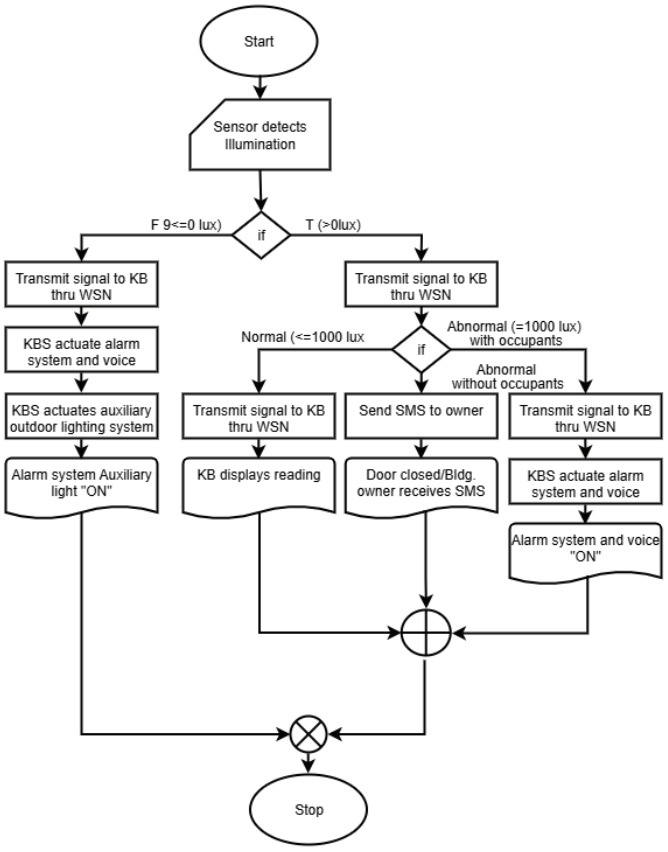


Figure B-8: Algorithm for Light Sensor Night Operation

http://dx.doi.org/10.21622/ACE.2026.06.1.1825

REFERENCES

[1] K. Amit, Artificial Intelligence and Soft Computing. Boca Raton , FL, USA: CRC Press, 2018. doi: 10.1201/9781315219738.

[2] Department of Information Technology, "Malla Reddy College of Engineering & Technology, Digital Notes on Artificial Intelligence." [Online]. Available: https://mrcet.com/downloads/digital_notes/IT/(R17A1204)%20Artificial%20Intelligence.pdf

[3] Arduino, "Getting Started with Arduino UNO." [Online]. Available: https://www.arduino.cc/en/Guide/ArduinoUno/

[4] M. R. Hayag, "ZIGBEE TECHNOLOGY UTILIZATION IN GATHERING AND TRANSCEIVING DATA FOR INTELLIGENT BUILDING AUTOMATION.," 2019.

[5] Arduino, "Industry Ready Solutions," 2026. [Online]. Available: https://www.arduino.cc/education/university/

[6] S. Cirani, G. Ferrari, M. Picone, and L. Veltri, Internet of Things: Architectures, Protocols and Standards. 2018. doi: 10.1002/9781119359715.

[7] "Stanford Artificial Intelligence Laboratory, AI Lab Brochure," 2018, Stanford University. [Online]. Available: https://ai.stanford.edu/

[8] Semaphore, "SMS Gateway Philippines - SMS API | Semaphore." [Online]. Available: https://semaphore.co/

[9] Tzone, "Tag11 LoRa Wireless Voltage and Current Transmitter Environment Monitoring System. ," 2016. [Online]. Available: https://www.tzonetemperature.com/

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

DESIGN AND EVALUATION OF A SOFTWARE-DEFINED NETWORKING INTRUSION DETECTION AND PREVENTION SYSTEM USING ENSEMBLE MACHINE LEARNING AND HYBRID FEATURE SELECTION

Olumhense Benedict Adoghe[1], Francis Emmanuel Kibuebu[1], Ejemen Comfort Ikpotokin[2] and Emmanuel Dominic Ephraim[3]

[1]Achievers University, Department of Electrical and Information Technology Engineering, Owo, Ondo State

[2]Auchi Polytechnic University, Basic Science Department, Auchi, Edo State, Nigeria

[3]Achievers University, Department of Biomedical Engineering, Owo, Ondo State, Nigeria

Emails: {adoghe.ob@achievers.edu.ng, kibuebuemmanuelf@gmail.com, comfortikpotokin@gmail.com, ephraim.ed@achievers.edu.ng}

Received 29 November 2025 - Accepted 23 January 2026 - Published 22 April 2026

ABSTRACT

Cyber threats in today's software-defined network infrastructures are growing at an exponential rate; thus, sophisticated, adaptive security measures are required. In order to enhance detection accuracy and computational efficiency, this study introduces a Network Intrusion Detection and Prevention System (NIDPS) that is ensemble-based and designed for Software-Defined Networking (SDN). The NIDPS makes use of a hybrid feature selection technique. As part of its soft-voting ensemble architecture, the system incorporates XGBoost, Decision Tree, and Support Vector Machine, three machine learning classifiers. To tackle the difficulties of dealing with high-dimensional network traffic data, feature selection is optimised using Correlation-Based methods and Recursive Feature Elimination (RFE). Datasets obtained from simulated Denial-of-Service (DoS) attacks were used to evaluate the model, which was constructed and tested in a virtualised, emulated SDN environment using the SEED Internet Emulator. All of the model variations demonstrated near-perfect detection ability (up to 100% accuracy) in the experiments, with the fastest predictions coming from RFE-enhanced models. The system is well-suited for implementation in actual programmable network settings due to its real-time alerting and preventive features. The results of this study show that an effective and scalable method for intrusion detection in SDNs may be achieved by combining ensemble learning with intelligent feature selection.

Keywords: *Network Intrusion Detection System (NIDS), Ensemble Machine Learning, Hybrid Feature Selection, Cybersecurity, Denial-of-Service (DoS) Attack Simulation*

INTRODUCTION

The radical change in data movement and network activity created by the groundbreaking influence of digital technology, cloud computing, and the IoT on modern communication infrastructures has led to a dramatic increase in data transfer and network activity [1, 2]. Due to the increase in cyberattacks, the number and destructive power of cyberattacks are rising, and because of these developments, novel issues in cybersecurity emerge [3]. Recent statistics show that network attacks will constitute a significant portion of the global damages of cybercrime that will wipe out over $10.5 trillion annually by 2025 [4, 5]. Intrusion Detection Systems (IDS) have become an invaluable security measure in this setting due to the burden of threat identification, mitigation, and prevention in real-time [6, 7]. Conventional methods of IDS, like signature-based ones, are based on given

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

attack patterns, and many are not able to respond to zero-day attacks or changes in threat vectors [8]. The same cannot be said about anomaly-based detection, where ML and DL are employed to identify suspected network activity and allow more latitude to detect a new threat [9]. Notwithstanding these advancements, the current IDS can yet achieve the objectives to overcome the challenges facing them, such as computational inefficiency, class imbalance, large false-positive rates (FPR), and high-dimensional data [10]. Recently, hybrid feature selection and ensemble machine learning models have been popular as solutions to these shortcomings [11]. These methods integrate the strengths of many algorithms to enhance the detection accuracy, generalisability, and robustness. This article explores how better network intrusion detection can be achieved by using enhanced ensemble ML models along with hybrid feature selection methods to draw a comprehensive overview of current methods, challenges, and possible future directions of the topic. The online transformation of industries, smart cities, and critical infrastructure exposes networks to more attacks since a larger attack surface is created. The Cybersecurity Ventures 2024 Report shows that one cyberattack happens every eleven seconds on average. Ransomware, Distributed Denial-of-Service (DDoS), and Advanced Persistent Threats (APTs) are considered to be the most prevalent forms of cyberattacks [12, 13]. Besides this, IoTID20 and CICIDS2017 evidence indicates that emerging threats have complicated patterns and regularly succeed in evading conventional security controls [14]. The issues with conventional approaches to intrusion detection systems (IDS) are that they have low generalisability due to their reliance on static attack signatures, high false-positive rates (FPR) resulting in false alarms causing unwarranted operational costs, and cannot support high-dimensional data owing to their inability to process large volumes of network traffic efficiently [15].

The constraints underscore the significance of smart intrusion detection systems (IDS) that are adaptive and can scale according to the needs of the modern cybersecurity threats. With the development of deep learning and machine learning, today, intrusion detection systems can automatically generate features, detect abnormalities, and classify threats in real-time [16]. With known attacks, RF, XgBoost, and neural networks can also be successfully applied in the case of supervised learning, whereas unsupervised (and semi-supervised) algorithms such as Autoencoders and LSTM are particularly effective when it comes to zero-day attacks [17, 18]. Recent research has demonstrated that hybrid DL models, such as CNN-LSTM and GRU-based models, can be very effective at capturing a sequence and spatial pattern of network traffic, as demonstrated in recent studies [17]. For instance, the XGBoost-LSTM hybrid model was capable of performing better than traditional methods with a 99.7 percent detection rate on the CICIDS2017 dataset [19]. Similarly, Deep Reinforcement Learning (DRL) has been employed in advancing the skills of detection by modifying the evolving methods of attack [20].

The biases in the models exist due to the class imbalance in which benign samples of traffic far exceed the amount of attack samples, making the models biased [9]; redundancy in features and noise, worsening the performance of the model and causing longer training sequences, are issues of concern [18, 21] that the ML/DL-based IDS continue to face. The challenges require the incorporation of sophisticated feature selection methods in order to maximize the efficiency and accuracy of the model. One of the most important preprocessing methods in IDS is feature selection, whose main purpose is to minimize dimensionality, remove irrelevant features, and enhance detection efficiency [22]. Conventional feature selection techniques can fall into either of the following categories: Filter Methods (e.g., Pearson Correlation, Chi-square, ANOVA) - Fast yet do not consider interactions between features 213, Wrapper Methods (e.g., Recursive Feature Elimination, Genetic Algorithms) - Slow but model-sensitive 3, and Embedded Methods (e.g., Lasso Regression, Random Forest Importance) - between efficiency and performance [23].

Recent investigation suggests that hybrid feature selection can increase intrusion detection performance by combining various feature selection strategies and capitalising on their individual strengths. To take advantage of the synergies between different feature selection paradigms, hybrid feature selection combines many approaches, most commonly filter and wrapper methods [8], [15]. Before training the model, this study uses correlation-based filtering to remove features that are highly correlated or redundant.

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

Then, it uses Recursive Feature Elimination (RFE) to find the features that are most discriminative. Improved computing economy without sacrificing classification accuracy is achieved using this hybrid technique [21]. As an illustration, MI-Boruta (Mutual Information + Boruta Algorithm) offered 99.9% accuracy on UNSW-NB15 with the combination of filter and wrapper, CFS-FPA (Correlation Feature Selection + Forest Penalized Attributes) had 0.004% false alarms on CICIDS2017, and the selection of features with XGBoost offered higher detection rates at lower training times of 27-63% [13], [24]. The advantages of these combined methods are not only the improvement of detection accuracy but also the guarantee of practical use in high-speed networks in real-time. Such methods as Bagging, Boosting, and Stacking of ensemble learning have become popular in IDS as they enable a combination of several weak classifiers into a formidable detector [14], [24]. Some of the strategies that are used as ensembles are Random Forest (RF) & XGBoost - Effective on binary and multi-class classification 9, AdaBoost and Gradient Boosting - Improve minority class detection in imbalanced datasets 3, and Stacked Ensemble Models (e.g., RF + XGBoost + MLP Meta-learner) - Achieve superior generalization [25]. Recent research findings have proven that hybrid feature selection stacked ensemble models are better than single classifiers. In the example, an RF-CatBoost-XGBoost model won 99.99% on NSL-KDD and CICIDS2017, and an AdaBoost-enhanced SVM lowered false negatives by half over classic SVM [26]. These results indicate the promise of ensemble-hybrid IDS models to overcome recent cybersecurity issues.

METHODOLOGY

This section of the research presents the approach adopted to conduct, design, develop, and implement an ensemble-based NIDPS within an SDN architecture.

STAGES OF PROJECT DEVELOPMENT

Here, the critical stages of the project execution are presented with a focus on the iterative process of design, development, implementation, and evaluation metrics to measure the efficiency of the system. Figure 1 below provides a stepwise methodology structure that is depicted in phases.

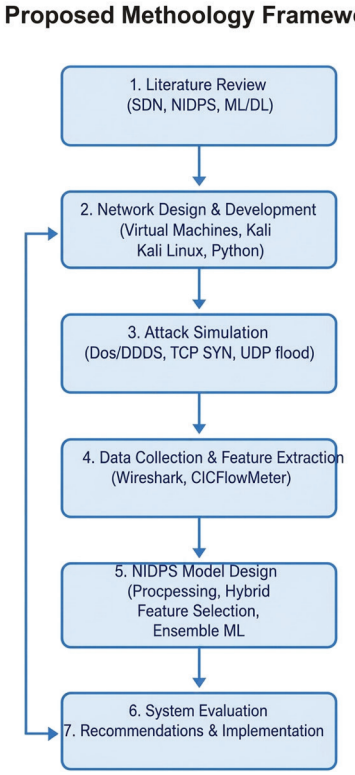


Figure 1: Proposed methodology framework

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

To ensure that the intrusion detection and prevention system is robust and effective, this study on Adopting Network Intrusion Detection by using the improved Ensemble Machine Learning Models with a Hybrid feature selection Model conforms to a process of steps. The initial one is to read the literature available on the topic at hand, which will provide you with an overview of such key technologies as Software-Defined Networking (SDN), NIDPS, and the past solutions. After this, a software-based network is designed and developed on virtual machines, and with the help of software such as Kali Linux, the network architecture is written and configured in Python. A virtual network is used to simulate Denial-of-Service (DoS) attacks, in which custom TCP SYN and UDP flood scripts are used to replicate the conditions of a real intrusion. During the data collection stage, Wireshark is used to capture network traffic of such attacks, and CICFlowMeter is used to extract features of the packet capture (.pcap) files and create machine learning-processable datasets. The second step is the development and design of an NIDPS model that takes into consideration the preprocessing of the obtained data, the use of hybrid feature selection algorithms to minimise dimensionality, and the governance of ensemble machine learning algorithms in order to increase detection accuracy. Lastly, the system is evaluated based on the performance in accordance with certain metrics to evaluate its efficiency, and the paper ends with recommendations for the implementation of the developed NIDPS system into practice, pointing to its practical importance and possible implications on the organizational cybersecurity practices.

DESIGN AND IMPLEMENTATION OF THE ENSEMBLE-BASED NIDPS

This part of the research paper presents the information about the design of the ensemble ML-based NIDPS that is integrated with an SDN architecture and tested on the performance measures. It starts with the network architecture design that illustrates the emulation of an advanced network, then the development and implementation of the NIDPS model in the isolated network. All the complex procedures that are entailed in this network design, the development of the NIDPS model, implementation, and evaluation are all reflected in this section. The proposed system has the overall network architecture as illustrated in Figure 2 below.

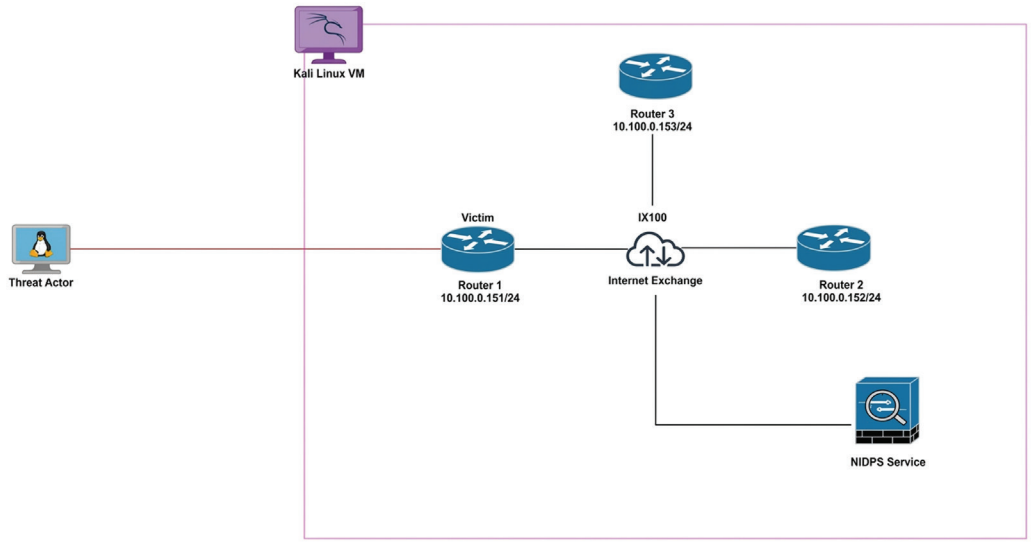


Figure 2: System Design and Architecture

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

VIRTUAL MACHINE (VM) SETUP AND CONFIGURATION

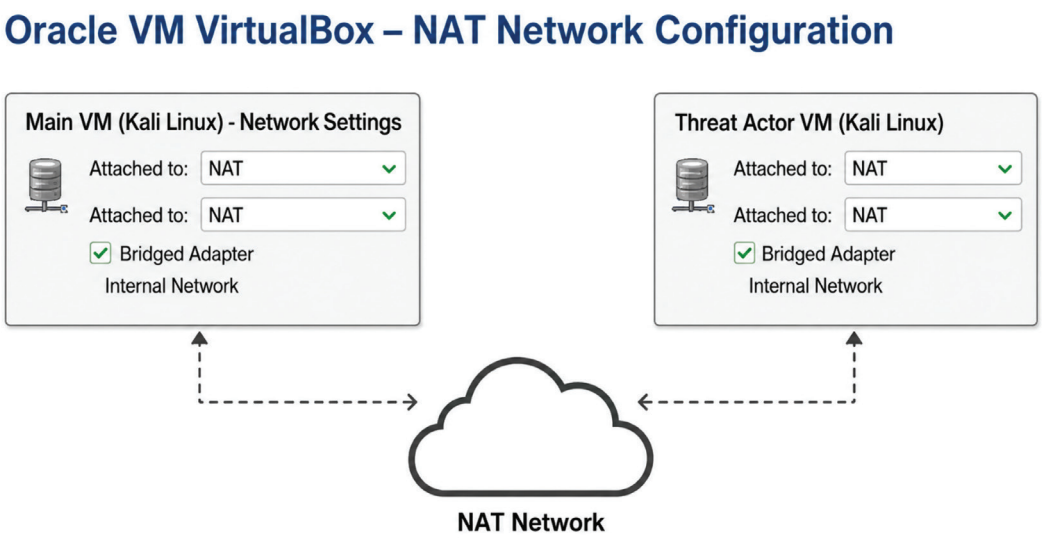


Figure 3: VM Setup and Configuration

In order to safely design an NIDPS, tested against simulated attacks, it was necessary to establish an isolated environment in which the methods of development and testing may be conducted. Therefore, Kali Linux OS was installed in the Oracle VM VirtualBox manager that was endowed with 4GB base memory, 2 processors, 80GB virtual space, and access to the internet was disabled to ensure that any accidental exposure of simulated threats to the internet did not occur. The details of this setup are indicated in Figure 3 below.

The second VM set up was performed as shown in Figure 4. This was the mechanism that the threat agent used in delivering DoS/DDoS attacks. Kali Linux OS was also installed in the VM, and it had 2GB of memory and 2 CPU processors. The threat actor-VM was then connected to the same network as the main VM, as shown in Figure 4, due to the configuration.

```
#!/usr/bin/env python3
# encoding: utf-8
from seedemu.layers import Base, Routing, Edge
from seedemu.services import Webservice
from seedemu.compiler import Docker
from seedemu.core import Emulator, Binding, Filter

# Function to create an autonomous system with a router connected to an IX
def setup_autonomous_system(base_layer, asn, router_name, ix_network):
    asys = base_layer.create_autonomous_system(asn)
    asys.createrouter(router_name).joinnetwork(ix_network)
    return asys

# Initialize the emulator and layers
emu = Emulator()
base = Base()
routing = Routing()
edge = Edge()

# Create an Internet Exchange
ix_number = 100
base.createInternetExchange(ix_number)

# Autonomous system numbers for the three ASes
as_numbers = [150, 351, 353]

# Setup autonomous systems
for asn in as_numbers:
    setup_autonomous_system(base, asn, f'router{asn - 150}', f'ix{ix_number}')

# Peering the ASes at Internet Exchange IX-100
for asn in as_numbers:
    edge.addaspeer(ix_number, asn)

# Add layers to the emulator and render
emu.addlayer(base)
emu.addlayer(routing)
emu.addlayer(edge)
emu.render()

# Compile the emulation environment
emu.compile(Docker(), './output')
```

Figure 4: Network Address Translation Configuration for both VMs.

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

NETWORK DEVELOPMENT

This was followed by the development of an emulated network on the SEED Internet Emulator (IE) software tool at this level, after having an isolated environment. The SEED IE provides a wide range of programmability of the network with configurable classes and modules that are written in Python. It is easy to simulate network behaviours using virtual network components, including routers, switches, and hosts, which are described as programmable classes. The network was coded to have three BGP routers that were linked by an internet exchange (IX), where each router was a representation of a local network that had web servers, user systems, and other services. The source code of the implementation and client interface of network management is presented in Figures 5 and 6, respectively.

Figure 5: Emulated Network Development.

AUTONOMOUS SYSTEMS AND INTERNET EXCHANGE POINT CONFIGURATION

As shown in Figure 6, the centre of the network configuration was the instantiation of Autonomous Systems (ASes), namely, asn_numbers, or, in this case, AS150, AS151, and AS152. These systems are isolated networks that have the ability to route independently. Each of the AS has one router, router0, router1, and router2, respectively, indicating the gateway on which the network traffic is controlled and shared. These routers were then linked together via a particular internet exchange point (IXP), which is referred to as ix100, as shown in Figure 6 below. The IXP is an important factor in determining whether the ASes are able to share the internet traffic.

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

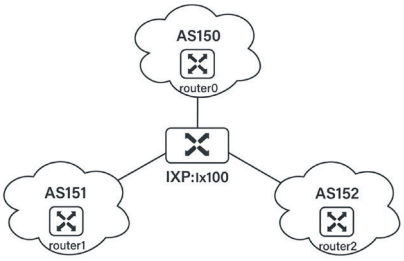


Figure 6: Autonomous Systems and Internet Exchange Point (IXP) configuration

BGP ROUTING IMPLEMENTATION

The IXP ASes were provided with the Border Gateway Protocol (BGP) peering sessions. This may be observed in Figure 7, with the following being the contents of the function: ebgp. RsPeer(ix number, asn). This facilitated the distribution of routing information throughout the network. This configuration directly reflects the working principle of internet routing and is directly related to the complex network architecture in which the exchange of information between ASes is to optimise path choice and traffic flow.

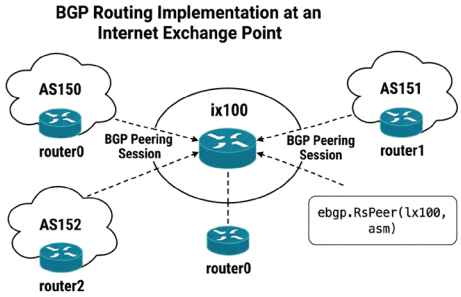


Figure 7: BGP Routing Implementation at an Internet Exchange Point

NETWORK MODULARITY

To simplify and modularise the process of constructing an Autonomous System (AS), a reusable configuration function was implemented. This function automates router configuration and establishes connectivity to the Internet Exchange Point (IXP), as illustrated in Figure 8.

```
#!/usr/bin/env python3
# encoding: utf-8
from seedemu.layers import Base, Routing, Ebgp
from seedemu.services import WebService
from seedemu.compiler import Docker
from seedemu.core import Emulator, Binding, Filter

# Function to create an autonomous system with a router connected to an IX
def setup_autonomous_system(base_layer, asn, router_name, ix_network):
    asys = base_layer.createAutonomousSystem(asn)
    asys.createRouter(router_name).joinNetwork(ix_network)
    return asys
```

Figure 8: Function for AS Configuration

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

NETWORK EXECUTION

Having established the various methods and the various network configurations, and the topology of the network, we used the compile method using the Docker argument, which is represented in Figure 9, to convert our abstract network topology to an executable emulation environment. The specified topology is transformed into a collection of interrelated Docker containers using this approach, with each of them executing one of the elements of a simulated network.

```
35    # Add layers to the emulator and render
36    emu.addlayer(base)
37    emu.addlayer(routing)
38    emu.addlayer(ebgp)
39    emu.render()
40
41    # Compile the emulation environment
42    emu.compile(Docker(), './output')
```

Figure 9: Emulated Network Compilation and Execution

Figure 10 presents the client interface of the network that is in compliance and is executed. Under this interface, we could get access to all the routers, visualise the connectivity, and the relationship of the network with the internet exchange point.

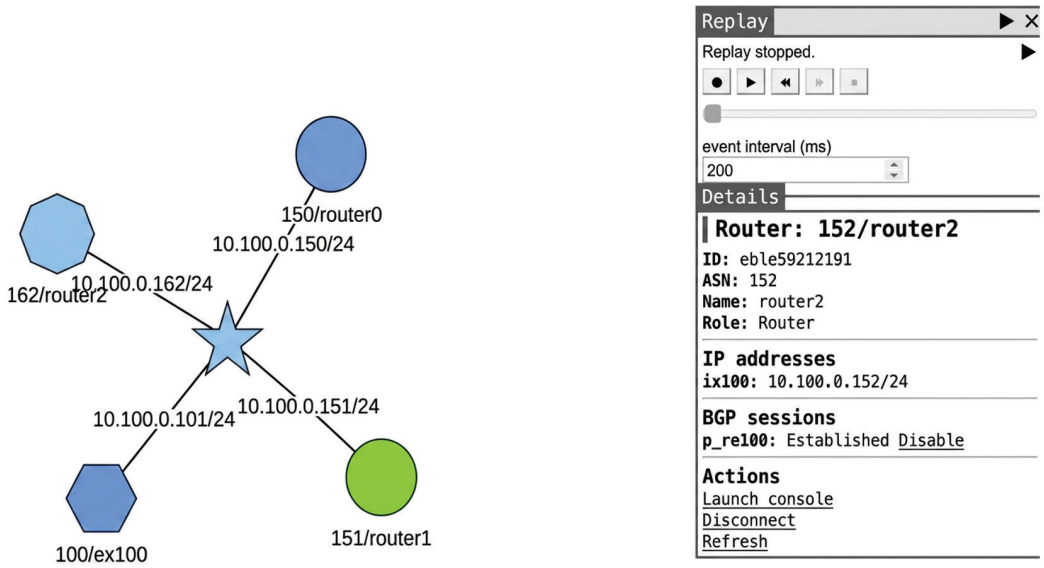


Figure 10: Client Interface for the Developed Emulated Network

DOS ATTACK SIMULATION

As our network environment was ready and operational, we created a sequence of DoS attacks that would be emulated to load stress on our network environment, and that would be utilized to test our proposed NIDPS model. The first one was a TCP SYN flood attack on a given IP address and port within the network. This attack is of a multithreaded nature, and it is an actual DDoS attack, which is a simulation, meaning that it floods the particular address with lots of packets, hoping to interfere with the normal functioning of the address. Figure 11 can be regarded as a logic of the packet configuration and thread sessions.

http://dx.doi.org/10.21622/ACE.2026.06.1.1834


```
// Configure Security
SecurityHttpConfig(httpSecurityConfig, jwtAuthFilter, authenticationManager, userDetailsService,
        tokenProvider, passwordEncoder, securityProperties, tokenCookie, tokenHeader);

class SymbolController {
   def __init__(self, target_up, target_port):
       super(SymbolController, self).__init__()
       self.target_up = target_up
       self.target_port = target_port

   def run(self):
      try:
            # Craft & fire exploit
            payload = Payload(self.target_up, f"{PayloadImpl}.template", Flag="1")
            # Define how to keep reading results
            while True:
                resultCacheE =VerboseWriter(
                   Logging.Output("resultFile", wait_sec_packet, fself.target_upself.target_port}")
       except Exception as ex:
            # Logging error/wrong sending (no aspects) {E}
            print(f"error: {ex}")

 def main(target_up, target_port, thread_count):
    threads = []
    for _ in range(thread_count):
        thread = Symbol(target_up, target_port)
        thread.start()
        threads.append(thread)
    for thread in threads:
        thread.join()

if __name__ == "__main__":
    target_up = input('Enter target IP: ')
```

Figure 11: TCP Syn Flood Attack.

Besides this, we went ahead to devise a UDP flood attack as in Figure 12, with the intention of flooding the network bandwidth, consuming and exhausting server resources. Here, we determined the size of the packets being sent, the threads, and the IP address and port of a particular IP.

- Packet Sending: Sending packets to a destination is performed by the script with a large number of threads running simultaneously. All threads flood packets to the limit of their capacity to the network and the target server.
- Packet Size: Packet size is set to 1024 bytes.
- Target Specification: The Packet is transmitted to a targeted IP address and port, which makes the attack focused and may be more disruptive.

```
THREAD_COUNT = 1000  # number of threads to use in the attack

def send_udp_packets():
    """
    Function to send UDP packets to the target and log each packet sent.
    """
    # Create a socket
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    # Generate random bytes to send
    bytes_to_send = random._urandom(PACKET_SIZE)

    while True:
        try:
            # Send the packet to the target
            sock.sendto(bytes_to_send, (TARGET_IP, TARGET_PORT))
            logging.info(f"Sent (PACKET_SIZE} bytes to {TARGET_IP):{TARGET_PORT}")
        except Exception as e:
            logging.error(f"Error: {e}")
            break

def main():
    """
    Main function to start the UDP flood attack and log the start and end of the attack.
    """
    logging.info(f"Starting UDP flood attack on {TARGET_IP):{TARGET_PORT} with {THREAD_COUNT}
    threads = []

    for _ in range(THREAD_COUNT):
        # Create a new thread to send UDP packets
        thread = threading.Thread(target=send_udp_packets)
        thread.start()
```

Figure 12: UDP Flood Attack

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

NIDPS MODEL DEVELOPMENT

In this part, we have presented detailed information about the various modules that have been created towards a complete intrusion detection and prevention system. The NIDPS will include the IDS module, where the threats are identified, which are recorded to be investigated by the network administrator. This was in turn succeeded by a long period of functionality, IPS, where the threats identified do not penetrate into the network. The last step of the model was to put in place an alert system that will alert the network administrator and a web interface to manage the system. The drawn sequence of activities illustrated in Figure 13 reflects simulated DoS attacks directed to the emulated network, the role of the NIDPS in the detection of the attack, and the timely alerting of the network administrator to take the required measures, which can be any of the following; blocking the IP addresses where the attacks originated, closing open ports and amending firewall rules.

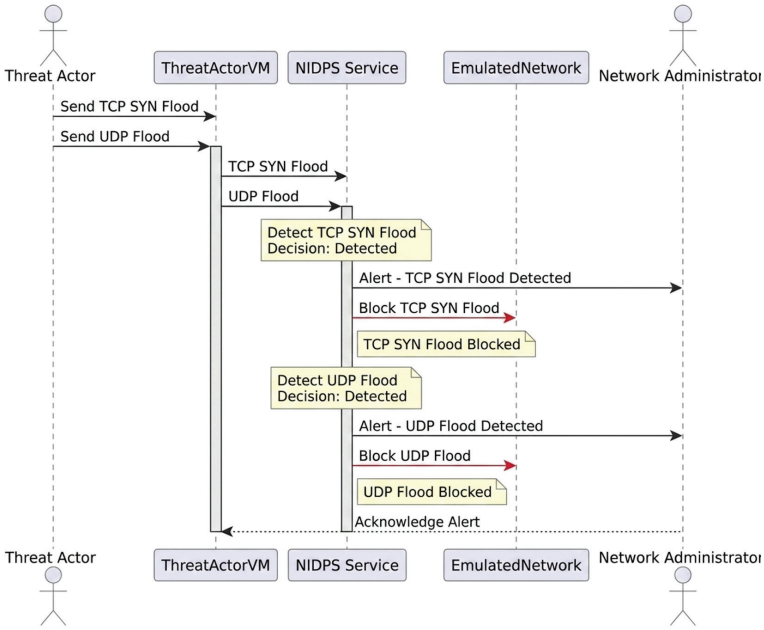


Figure 13: DoS Attack and NIDPS Sequence Diagram

NIDPS MODULE

To have the IDS module work as expected, we have gathered packet traffic data on the basis of simulated normal traffic and DoS attack traffic that was captured using the Wireshark tool and a custom-built Python-based packet sniffer script. The script also made use of the library known as PyShark, which captured and analysed network traffic. As has been mentioned above, we simulated two DoS security attacks, tcpsynflood.py and udpflood.py. Features were extracted by the packet sniffer and analysers, depending on both simulated attacks in the CICFlowmeter open source application. The resulting records were abridged into TCPSYNFloodDataset.csv and UD PfloodDataset.csv, respectively. The table and graphical distribution of the types of attack are as illustrated in Table 1 and Figure 14, respectively.

Table 1: Dataset Distribution of Traffic Types

Traffic Type	Count
Normal	7,000
TCP SYN Flood	5,496
UDP Flood	5,090

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

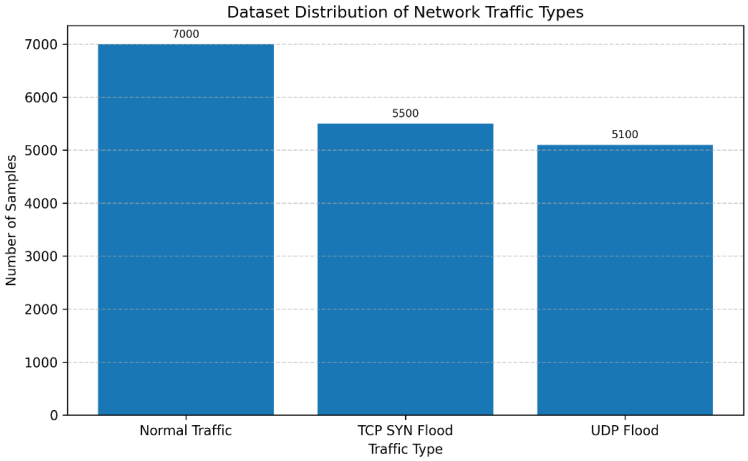


Figure 14: Dataset Distribution

The second stage of development implied cleaning the features that had been extracted and preprocessing them with pre-defined functions and libraries. A column with a label was added to each dataset that contained the normal traffic dataset, and empty fields were deleted. This was in order to embrace a supervised learning method of the ML model. We have continued with the label encoding process wherein we called the LabelEncoder() module to encode the categorical data into numerical ones. Since the packet sniffing and feature extraction steps were already done, the dataset generation and preprocessing steps were also done. The following step in the development of the NIDPS was to train and test the ensemble ML models using the dataset. One of the most important steps of the model development was the feature selection, as it is important to determine the important and relevant features to be used to train the model. Firstly, the Recursive Feature Elimination with Cross Validation (RFECV) method was embraced in order to determine significant features to be included in the training. This method of feature selection applies random forest as an estimator that recursively adopts the features in order to decide the relevance of the features using a set of pre-defined characteristics. In this step, cross-validation goes on to remove irrelevant features and minimise data noise. As far as data classification in IDS is concerned, RFECV boasts of improved performance (Nguyen, 2022). The RFECV was capable of identifying and reducing the number of features to be used in training, among 82 features, to 10 features. Figure 15 is the importance scale of each feature. Moreover, an experimental comparison was adopted with another feature selection process. The Correlation-based Feature Selection method in this case will pick the subset of features that will give more information. The method that has been found to enhance the performance of classifiers such as DT and SVM (Farahani, 2020) is based on statistical correlation among features that eliminates redundant features. Through this, 47 features were picked among 82 features that had the least interdependence with each other. The correlation of features is given in Figure 16.

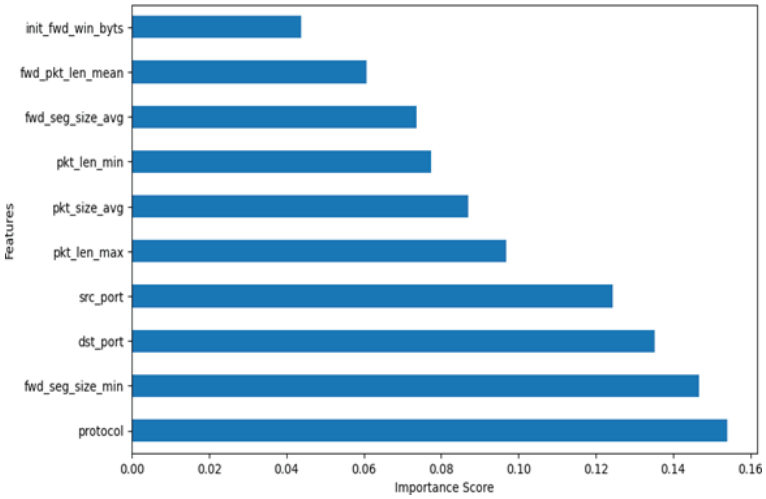


Figure 15. Feature Importances of Selected Features by RFECV

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

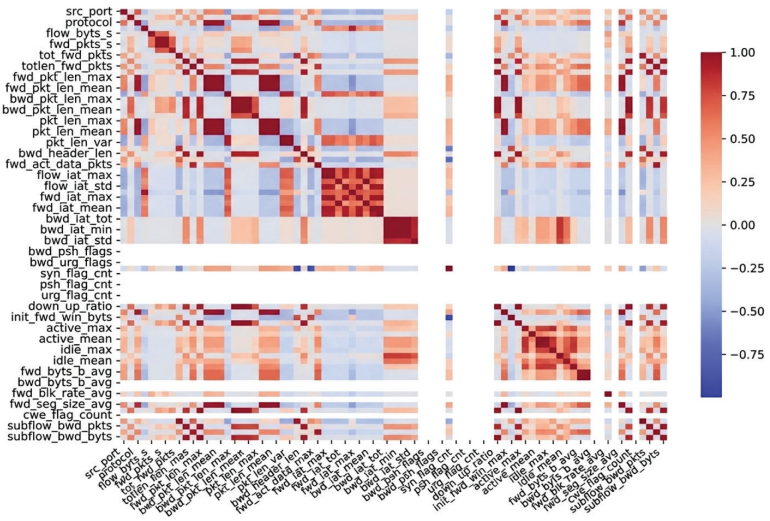


Figure 16. Correlation Matrix of Features

After applying feature selection techniques and identifying features to use with both methods, DT, XGBoost, and SVM classifiers were trained and tested individually, and finally, the voting ensemble consisting of the three classifiers was used to combine their predictive value in identifying malicious and benign traffic. Figure 17 shows the ordered diagram of the NIDPS model, beginning with the packet sniffing process until the deployment of the model. Another facility was implemented on the model, and that is the email alert generation and the IPS module. Using these, when the NIDPS identifies malicious traffic across the network, it will create an email message that will be sent to the network administrator, and the network administrator is then allowed to invoke the IPS feature. This gave the administrator authority to view the specifics of the network traffic and determine the course of action they can take, either to block or permit.

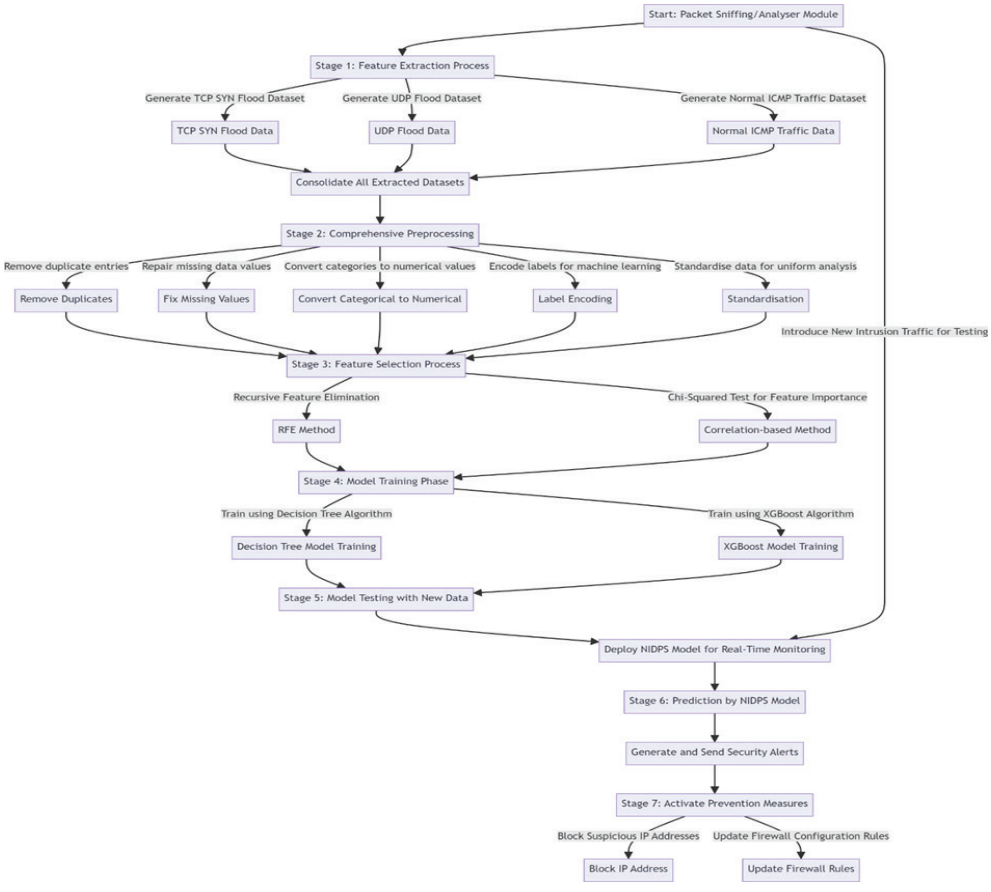


Figure 17: NIDPS Model Training and Operation Architecture

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

SECURITY ALERT GENERATION

This stage of the model is when the model has forecasted an intrusion following its strict training and testing. The alert generation functionality sends an email notification to the network administrator of an intrusion and a brief perspective of what kind of attack it is, the time of attack, the source and target IP addresses, and the destination port number. This provides the administrator with a short understanding of the intrusion and can take the required measures where necessary. Fig. 18 is the email reply that the administrator will receive when the model predicts that there is an intrusion in the network.

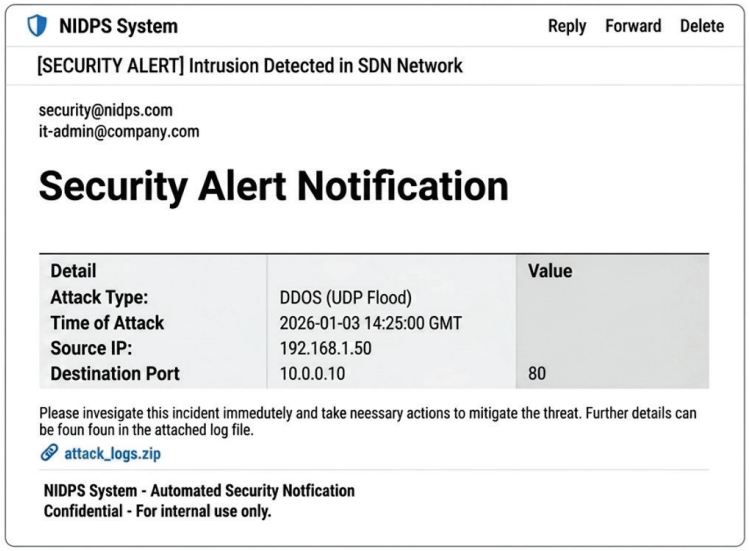


Figure 18: Security Alert Notification

3.0 RESULTS

In this part of the research paper, a close analysis of the findings and an assessment of the suggested NIDPS model will be given. Our model has been developed on the basis of three big experiments. Our model was an ensemble of three ML classifiers (DT, XGBoost, and SVM). To assess the NIDPS model, a number of metrics were taken into consideration, particularly in light of the importance of reducing false positives and the advanced speed of detecting real intrusions within the network.

A. Precision: This measures the accuracy of the positive predictions made by the model; it essentially quantifies the number of true positive results in the context of all positive predictions made (including both true positives and false positives). This metric can be represented in a mathematical formula as seen below:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \times 100\ \%$$

B. Accuracy: This represents the overall correctness of the model. This is calculated as the ratio of correctly predicted observations (both true positives and true negatives) to all observations in the dataset. It can be mathematically expressed as seen below:

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negative}}{\text{Total number of observations(samples)}}$$

C. Recall: Recall, or sensitivity, measures the model's ability to correctly identify all actual positives from the data, thereby assessing the model's capability to find all relevant instances. This is particularly critical to identifying actual threats, reducing the possibility of missing threats to a negligible level. The mathematical representation can be seen below.

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positive + False Negative}} \times 100\ \%$$

D. F1-Score (F1-S): This provides a balance between Precision and Recall, which is particularly beneficial in situations where there is an imbalanced data distribution. It is the harmonic mean of Precision and Recall.

$$\text{Recall} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision + Recall}} \times 100\ \%$$

In addition to these metrics, system throughput and resource usage are also measured to determine the impact of the implementation of the NIDPS on the network, as the aim is to ensure network performance is not significantly traded off for network security.

ENSEMBLE-BASED NIDPS WITHOUT FEATURE SELECTION

According to section 4, 82 features were picked out of the data file (.pcap file) of network traffic information received using the packet sniffer module of the NIDPS. These 82 features were directly used in training the individual base classifiers of the ensemble model, with all of them having considerably high training and testing accuracy as observed in Table 2. We went on to apply the voting ensemble that uses the majority voting method, whereby it capitalizes on the strong aspects of the base classifiers and offers a tradeoff of their weak aspects to improve model prediction. Under this ensemble method, we took a mild voting strategy. This is achieved by using the maximum prediction probability of the base classifiers. Table 2 shows that the voting ensemble obtained a perfect score on all metrics with no feature selection method implemented, whereas Figure 19 provides a confusion matrix associated with this approach. Furthermore, the time taken by the ensemble model to make predictions was 0.065 seconds, as observed in Table 5.

Table 2: Performance Evaluation of the Ensemble in Comparison with Each Classifier

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	1.0000	1.0000	1.0000	1.0000
XGBoost	1.0000	1.0000	1.0000	1.0000
SVM	0.9994	0.9994	0.9994	0.9994
Ensemble	1.0000	1.0000	1.0000	1.0000

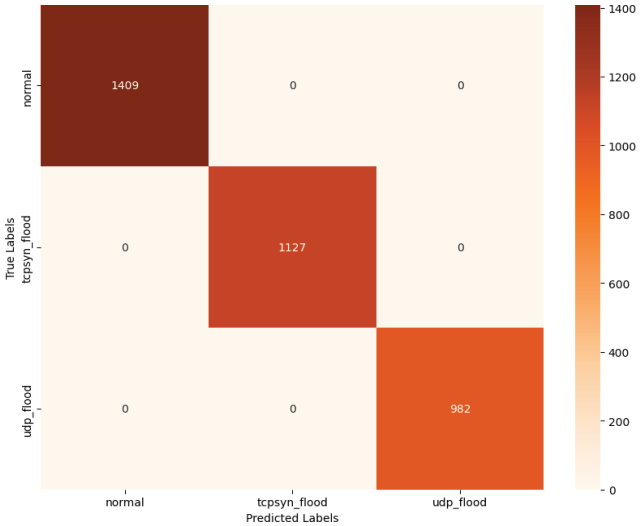


Figure 19: Confusion Matrix for the Ensemble Model without Feature Selection Method

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

ENSEMBLE-BASED NIDPS WITH RFE

The ensemble-based NIDPS with RFE made use of 10 features picked by the dataset using the recursive feature elimination correlation validation feature selection procedure. In this method of selecting features, the random forest classifier is used. The method was also producing high results in the various performance measures, such as precision, accuracy, recall, and F1-score, that are displayed in Table 3. Moreover, the value of each feature that is significant in the model prediction can be illustrated visually in Figure 20. The confusion matrix in Figure 17 also gives more insights into this technique. The ensemble model that used RFE also predicted the test data in 0.0358 seconds, as shown in Table 5.

Table 3: Performance Evaluation of the Ensemble Model and Standalone Classifiers with RFE

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	1.0000	1.0000	1.0000	1.0000
XGBoost	1.0000	1.0000	1.0000	1.0000
SVM	1.0000	1.0000	1.0000	1.0000
Ensemble	1.0000	1.0000	1.0000	1.0000

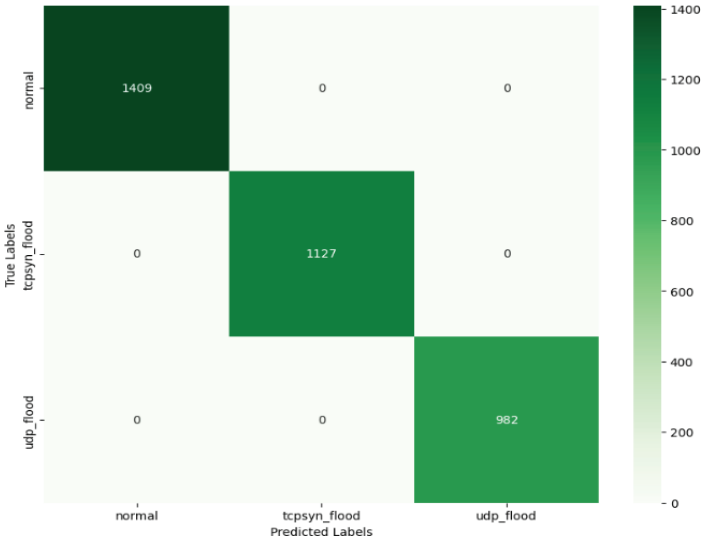


Figure 20: Confusion Matrix for the Ensemble Model with the Recursive Feature Elimination Method

ENSEMBLE-BASED NIDPS WITH CORRELATION-BASED FEATURE SELECTION

The feature selection method was also the correlation-based method, which was used to further test the performance of the ensemble model against the other two model implementation methods. The feature interdependency threshold was 0.9, which was a high threshold. This threshold allowed us to filter out features that are more interdependent by only selecting those features that are below the threshold, and we were left with 47 statistically unique features. The second step was to continue with the ensemble model training with the chosen features, and once again, the performance of the model was excellent, with all the classifiers having a 100 percent accuracy except the SVM with 99.94 percent, as observed in Table 4. The confusion matrix in Figure 21 provides more information on the matter, whereas Table 4 illustrates the various measures of performance. Also, the speed of prediction on this model was taken into consideration, as observed in the ensemble comparison table in Table 5.

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

Table 4: Performance Evaluation of the Ensemble Model and Standalone Classifiers with Correlation-Based Feature Selection

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	1.0000	1.0000	1.0000	1.0000
XGBoost	1.0000	1.0000	1.0000	1.0000
SVM	0.9994	0.9994	0.9994	0.9994
Ensemble	1.0000	1.0000	1.0000	1.0000

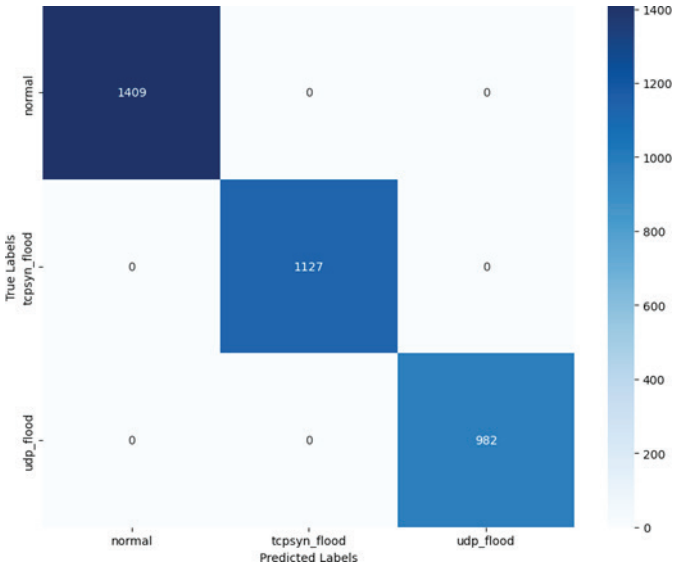


Figure 21: Confusion Matrix for the Ensemble Model with Correlation-Based Feature Selection Method

PERFORMANCE COMPARISON OF THE ENSEMBLE MODELS

Since the various ensemble models have been applied with and without the use of various feature selection methods, it is necessary and urgent to test the effectiveness of each of the models applied. We have not only compared the general performance measures of each model in our analysis, but have also noted the computational efficiency of the models, particularly the speed of prediction. This is an evaluation that is presented in Table 5, which gives an in-depth perspective of the NIDPS ensemble model performances. Besides that, figures 22 and 23 demonstrate the graphic depiction of such performance comparisons. The comparisons provide an insight into the comprehension of what models provide the best tradeoff between detection performances and the ability to operate, particularly in the case of implementation in a real-time network environment.

Table 5: Performance Comparison of the Ensemble Models in relation to Computational Speed

Model	Accuracy	Precision	Recall	F1 Score	Prediction Speed
Ensemble Model without Feature Selection	1.0000	1.0000	1.0000	1.0000	0.0755
Ensemble Model with RFE	1.0000	1.0000	1.0000	1.0000	0.0358
Ensemble Model with Correlation-Based Feature Selection	1.0000	1.0000	1.0000	1.0000	0.127

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

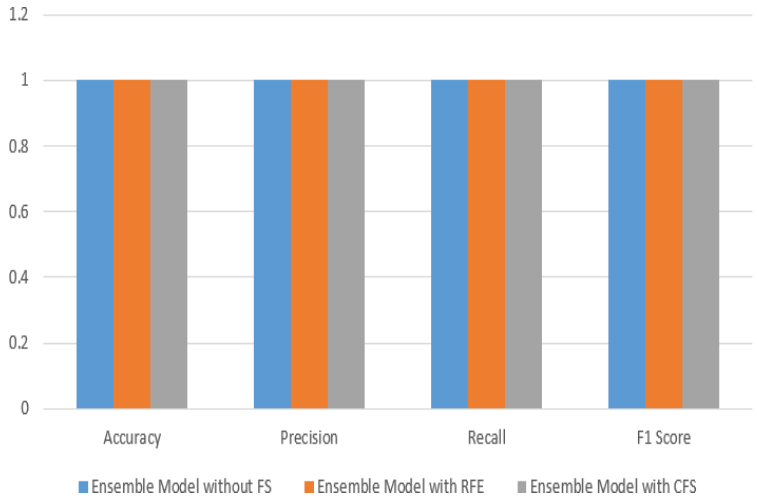


Figure 22: Performance Metrics Comparison for the Ensemble Models

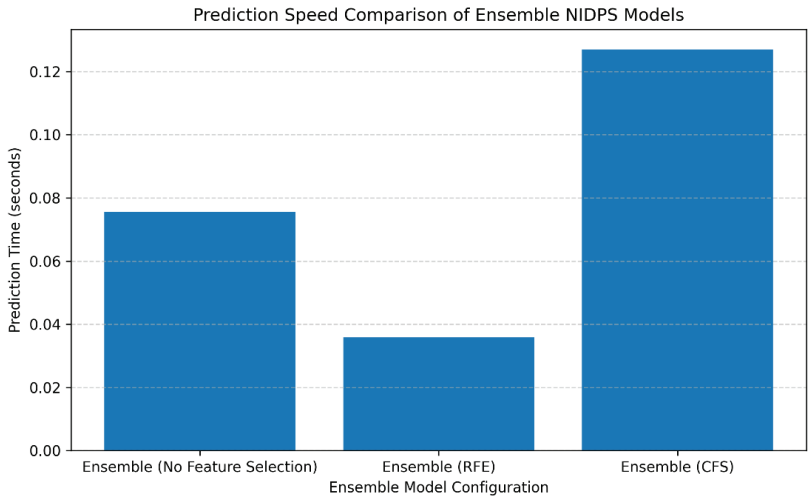


Figure 23: Prediction Speed for each Ensemble Model Implementation

DISCUSSION

The ensemble-based NIDPS system proposed in this paper demonstrated a high level of detection during all of the experiments performed in three experimental environments: no feature selection, Recursive Feature Elimination (RFE), and Correlation-Based Feature Selection. While Denial-of-Service (DoS) attacks were the main focus of this study, intrusion detection systems in real-world situations encounter a far broader range of attacks, including probing, privilege escalation, malware propagation, and zero-day threats. Due to its reproducibility, impact on traffic, and suitability for controlled experimentation in an SDN context, denial-of-service (DoS) attacks were our primary emphasis instead of attempting to capture every possible sort of attack. Having an almost zero difference in the recall and precision of a classifier, all ensemble variants achieved a 100 percent accuracy, which shows the versatility and stability of ensemble learning in a cybersecurity environment.

A. Ensemble Model with no Feature Selection

The ensemble model without feature selection showed the best classification performance, with all accuracy, precision, recall, and F1 score equal to 1.000, which shows that the combination of the classifiers (Decision Tree, XGBoost, and SVM) was able to learn using the full set of 82 features. However, the crude model had a relatively high prediction latency (0.0755 seconds), which can limit its use in real-time network applications, especially when the network is experiencing heavy traffic.

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

B. Recursive Feature Elimination (RFE)

The usage of RFE, a wrapper-based feature selection approach, reduced the number of features by 10 important ones and maintained the same classification accuracy. The RFE-based model had superior computational performance, and its fastest rate of prediction was 0.0358 seconds. This shows that RFE can be useful in reducing dimensions and improving performance, which is in line with earlier studies that point to the ability of the method to enhance the generalisability of a model and efficiency in high-dimensional data scenarios.

C. Correlation-Based Feature Selection

The third experimental setting, which featured Correlation-Based Feature Selection, also gave excellent results. This arrangement was a little longer than predicted (0.127 seconds). This method gave a satisfactory tradeoff between precision and a middle range shrinkage of features, although it retained 47 features (an artifact of its focus on reducing redundancy instead of aggressively dimensionality reduction). The reduced feature interdependence supported model interpretability and resilience, especially when dealing with the multicollinear effects of network traffic data.

KEY FINDINGS

These findings, combined with each other, bear witness to some significant observations:

A. Reliability of Models Using Different Feature Approaches

The integration of several classifiers enhances the limits of decisions and offsets the shortcomings of the single models. The ensemble model was more robust in the detection and classification of DoS attacks, independent of the method of feature selection.

B. Feature Selection Tradeoffs

Even though each of the methods was very precise, RFE offered the best compromise of speed and computing economy. This tradeoff is particularly crucial in environments where the resources are few, such as SDNs founded on the Internet of Things or on edge networks.

C. Real-Time Readiness

The latency of all variations of NIDPS is very low, and it can be used in real-time. This system will be best suited in a live network environment as it can be used to support administrative decisions by providing integrated IPS capabilities, as well as raising alerts at the appropriate time.

D. Scalability and Emulation Effectiveness

The validity of the scalability and operational reliability of the model in the face of controlled DoS attacks was tested by using the SEED internet emulator to recreate a distributed network environment in a realistic way. This is further supported by the adoption of enterprise programmable networks at scale.

LIMITATIONS AND FUTURE DIRECTIONS

The encouraging results have a number of caveats that must be taken into consideration. The experiments were first conducted on a simulated controlled dataset; to further generalise, it would be required to conduct the experiments on other forms of traffic, e.g., benign outliers and more complex multi-stage attacks. Secondly, soft vote aggregation enhanced the consensus of classifiers, but fine-tuning prediction could be enhanced with further research on dynamic ensemble optimisation systems such as meta-learning or adaptive boosting in the face of an evolving threat. In order to make the recommended NIDPS more representative of real-world network situations, further research will include evaluating it against more types of attacks and using datasets with significant imbalances.

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

CONCLUSION

We have described the process of designing and evaluating a Network Intrusion Detection and Prevention System (NIDPS) in this study that employs an ensemble strategy and would run within an SDN environment. It was demonstrated that the proposed system was highly sensitive to detecting Denial-of-Service (DoS) attacks, computationally efficient, and responsive in real-time because it employed a hybrid method that integrates different machine learning classifiers - Decision Tree, XGBoost, and Support Vector Machine - with sophisticated feature selection methods. The goal of all three experiments was to reach almost perfect or ideal detection values, and the result is what it accomplished using full features, Recursive Feature Elimination (RFE), and Correlation-Based Feature Selection. In the context of real-time network monitoring schemes, however, the ensemble model based on RFE really came into its own owing to its blistering prompt responses. We find that we can achieve increased scalability, responsiveness, and detection in the system by using a convergence of feature optimisation and ensemble learning. Simulation of the architecture on real-life network infrastructures and attack scenarios with the SEED Internet Emulator demonstrated the practicality of the suggested architecture. Moreover, the real-time production of alert features, as well as preventive functionality, is incorporated into the problem response process that gives network managers a response to an action. Its strengths notwithstanding, the current model has primarily been applied to controlled DoS traffic. The future research objectives should be to evaluate the system against a broader range of attacks, such as zero-day threats, in addition to putting the research through its paces in real-world SDN testbeds to understand how it will respond to changes over time. Other potential improvements are the integration of deep learning, federated learning in the case of distributed context, and dynamic reconfiguring of ensembles.

REFERENCES

[1] H. LI, Q. LI, Z. XU, and X. YE, "Digital Technologies," *Journal of Digital Economy*, vol. 3, pp. 240–248, Mar. 2025, doi: https://doi.org/10.1016/j.jdec.2025.02.001.

[2] O. B. Adoghe, B. A. Ikharo, H. E. Amhenrior, J. Abiodun, and E. B. Chukwu, "Data Collection Module for a Centralized Electronic Health Records System," *Journal of Engineering Research Innovation and Scientific Development (JERISD)*, vol. 2, no. 1, pp. 11–19, Mar. 2024, doi: https://doi.org/10.61448/jerisd21242.

[3] R. Kaur, D. Gabrijelčič, and T. Klobučar, "Artificial intelligence for cybersecurity: Literature review and future research directions," *Information Fusion*, vol. 97, no. 101804, pp. 1–29, 2023, doi: https://doi.org/10.1016/j.inffus.2023.101804.

[4] R. Muggah and M. Margolis, "Why we need global rules to crack down on cybercrime," *World Economic Forum*, Jan. 02, 2023. https://www.weforum.org/stories/2023/01/global-rules-crack-down-cybercrime/ (accessed Apr. 22, 2025).

[5] S. Morgan, "Cybercrime to cost the world $10.5 trillion annually by 2025," *Cybercrime Magazine*, Nov. 13, 2020. https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/ (accessed Apr. 22, 2025).

[6] L. Diana, P. Dini, and D. Paolini, "Overview on Intrusion Detection Systems for Computers Networking Security," *Computers*, vol. 14, no. 3, p. 87, Mar. 2025, doi: https://doi.org/10.3390/computers14030087.

[7] M. Rahman, S. A. Shakil, and M. R. Mustakim, "A Survey on Intrusion Detection System in IoT Networks," *Cyber Security and Applications*, p. 100082, Dec. 2024, doi: https://doi.org/10.1016/j.csa.2024.100082.

[8] O. Joseph and R. Ajax, "Comparison of Traditional vs. AI-Based Intrusion Detection and Prevention Systems: Efficiency and Accuracy," *ResearchGate*, Mar. 10,

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

2025. https://www.researchgate.net/publication/389717300_Comparison_of_Traditional_vs_AI-Based_Intrusion_Detection_and_Prevention_Systems_Efficiency_and_Accuracy (accessed Apr. 22, 2025).

[9] H. Dhrir, M. Charfeddine, N. Tarhouni, and H. M. Kammoun, "Machine learning- and deep learning-based anomaly detection in firewalls: a survey," *The Journal of Supercomputing*, vol. 81, no. 6, Apr. 2025, doi: https://doi.org/10.1007/s11227-025-07212-y.

[10] S. Muneer, U. Farooq, A. Athar, M. A. Raza, T. M. Ghazal, and S. Sakib, "A Critical Review of Artificial Intelligence Based Approaches in Intrusion Detection: A Comprehensive Analysis," *Journal of engineering*, vol. 2024, pp. 1–16, Apr. 2024, doi: https://doi.org/10.1155/2024/3909173.

[11] T. Tong and Z. Li, "Predicting learning achievement using ensemble learning with result explanation," *PLoS ONE*, vol. 20, no. 1, pp. e0312124–e0312124, Jan. 2025, doi: https://doi.org/10.1371/journal.pone.0312124.

[12] Kristel, A.-F. Rutkowski, and C. Saunders, "The whole of cyber defense: Syncing practice and theory," *The Journal of Strategic Information Systems*, vol. 33, no. 4, pp. 101861–101861, Sep. 2024, doi: https://doi.org/10.1016/j.jsis.2024.101861.

[13] N. James, "How Many Cyber Attacks Per Day: The Latest Stats and Impacts in 2023 - Astra Security Blog," *ASTRA*, Apr. 18, 2023. https://www.getastra.com/blog/security-audit/how-many-cyber-attacks-per-day/ (accessed Apr. 22, 2025).

[14] C. Rajathi and P. Rukmani, "Hybrid Learning Model for intrusion detection system: A combination of parametric and non-parametric classifiers," *Alexandria Engineering Journal*, vol. 112, pp. 384–396, Nov. 2024, doi: https://doi.org/10.1016/j.aej.2024.10.101.

[15] K. Noor, Agbotiname Lucky Imoize, C.-T. Li, and C.-Y. Weng, "A Review of Machine Learning and Transfer Learning Strategies for Intrusion Detection Systems in 5G and Beyond," *Mathematics*, vol. 13, no. 7, pp. 1088–1088, Mar. 2025, doi: https://doi.org/10.3390/math13071088.

[16] T. Sowmya and E.A. Mary Anita, "A comprehensive review of AI based intrusion detection system," *ScienceDirect*, vol. 28, pp. 100827–100827, Jun. 2023, doi: https://doi.org/10.1016/j.measen.2023.100827.

[17] P. A. doost, S. S. Moghadam, E. Khezri, A. Basem, and M. Trik, "A new intrusion detection method using ensemble classification and feature selection," *Scientific Reports*, vol. 15, no. 1, Apr. 2025, doi: https://doi.org/10.1038/s41598-025-98604-w.

[18] U. Ahmed *et al.*, "Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering," *Scientific Reports*, vol. 15, no. 1, Jan. 2025, doi: https://doi.org/10.1038/s41598-025-85866-7.

[19] S. L. Pawar and M. S. Karyakarte, "A Hybrid CNN and Attentive Hierarchical BiLSTM Model with SMO for Intrusion Detection in IIoT," Apr. 2025, doi: https://doi.org/10.21203/rs.3.rs-6158243/v1.

[20] J. Xie, "Application Study on the Reinforcement Learning Strategies in the Network Awareness Risk Perception and Prevention,"*The International journal of computational intelligence systems/International journal of computational intelligence systems*, vol. 17, no. 1, May 2024, doi: https://doi.org/10.1007/s44196-024-00492-x.

[21] B. Susilo, A. Muis, and R. F. Sari, "Intelligent Intrusion Detection System Against Various Attacks Based on a Hybrid Deep Learning Algorithm," *Sensors*, vol. 25, no.

http://dx.doi.org/10.21622/ACE.2026.06.1.1834

2, pp. 580–580, Jan. 2025, doi: https://doi.org/10.3390/s25020580.

[22] S. Walling and S. Lodh, "Enhancing IoT intrusion detection through machine learning with AN-SFS: a novel approach to high performing adaptive feature selection," *Discover Internet of Things*, vol. 4, no. 1, Oct. 2024, doi: https://doi.org/10.1007/s43926-024-00074-5.

[23] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr, and J. M. O'Sullivan, "A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction," *Frontiers in Bioinformatics*, vol. 2, Jun. 2022, doi: https://doi.org/10.3389/fbinf.2022.927312.

[24] N. Anand, R. Sehgal, S. Anand, and A. Kaushik, "Feature selection on educational data using Boruta algorithm," *International Journal of Computational Intelligence Studies*, vol. 10, no. 1, p. 27, 2021, doi: https://doi.org/10.1504/ijcistudies.2021.113826.

[25] A. A. Khan, O. Chaudhari, and R. Chandra, "A Review of Ensemble Learning and Data Augmentation Models for Class Imbalanced problems: Combination, Implementation and Evaluation," *Expert Systems with Applications*, vol. 244, no. 122778, p. 122778, Dec. 2023, doi: https://doi.org/10.1016/j.eswa.2023.122778.

[26] A. M. Alsaffar, M. Nouri-Baygi, and H. M. Zolbanin, "Shielding networks: enhancing intrusion detection with hybrid feature selection and stack ensemble learning," *Journal of Big Data*, vol. 11, no. 1, Sep. 2024, doi: https://doi.org/10.1186/s40537-024-00994-7.

[27] H. Duy, "Recursive Feature Elimination Algorithm for Intrusion Detection Systems," *2022 7th International Conference on Business and Industrial Research (ICBIR)*, May 2022, doi: https://doi.org/10.1109/icbir54589.2022.9786411.

[28] A. Fausto, G. Gaggero, F. Patrone, and M. Marchese, "Reduction of the Delays Within an Intrusion Detection System (IDS) Based on Software Defined Networking (SDN)," *IEEE Access*, vol. 10, pp. 109850–109862, 2022, doi: https://doi.org/10.1109/access.2022.3214974.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

ETHEREUM FRAUD DETECTION USING SMOTE-TOMEK AND STACKED ENSEMBLE LEARNING TECHNIQUE

Jamiu Muhammed Usman[1], Muhammad Nazeer Musa[2], Mario Kolberg[3], Nafisat Abdulkadir[4] and Habib Mohammed[5]

[1]University of Stirling, United Kingdom

[2]NIGERIAN DEFENCE ACADEMY, KADUNA, NIGERIA

[3]University of Stirling, United Kingdom

[4]Confluence University of Science and Technology, Osara, Nigeria

[5]North Carolina State University, Raleigh, USA

Emails: {jmu00001@students.stir.ac.uk, muhammadmusa2502@nda.edu.ng, mario.kolberg@stir.ac.uk, abdulkadirn@custech.edu.ng, himohamm@ncsu.edu}

Received 29 December 2025 - Accepted 02 February 2026 - Published 26 April 2026

ABSTRACT

This research addresses the critical challenge of detecting fraudulent Ethereum transactions, which remains notoriously difficult due to the overwhelming prevalence of legitimate activities compared to illicit ones. This pronounced class disparity frequently results in skewed outcomes when employing conventional detection frameworks. Our investigation implements a composite resampling strategy, SMOTE-Tomek, which concurrently augments the under-represented fraud category whilst eliminating ambiguous instances from the dominant category. Subsequently, the equalised dataset undergoes processing via a stacked ensemble framework comprising three robust gradient boosting methodologies: XGBoost, LightGBM, and CatBoost, synthesised by a logistic regression meta-classifier. Through stratified 5-fold cross-validation conducted against established benchmarks, the proposed ensemble attained a mean accuracy of 0.9873, whilst precision, recall, and F1-score each achieved an exceptional score of 0.98. Benchmark comparisons verified that our stacked methodology consistently surpasses individual base algorithms. The study demonstrated that combining targeted resampling with ensemble stacking offers highly effective solutions for Ethereum fraud detection, though real-time implementation and cross-blockchain generalisability require further investigation. The proposed model provides cryptocurrency exchanges, investors, and regulatory bodies with a robust mechanism for identifying fraudulent activities, thereby enhancing ecosystem security and maintaining user confidence. This work represents the first comprehensive integration of SMOTE-Tomek resampling with stacked gradient boosting ensembles for Ethereum fraud detection, validated through rigorous k-fold cross-validation rather than simple train-test splits.

Keywords: *Ethereum Fraud Detection, XGBoost, LightGBM, Stacking Ensemble, SMOTE-Tomek.*

1. INTRODUCTION

Since its establishment by Vitalik Buterin in 2015, Ethereum has distinguished itself as a prominent blockchain platform, largely attributed to its groundbreaking deployment of smart contracts and autonomous agreements with predetermined conditions encoded directly [1]. These programmable contracts have fundamentally transformed digital commerce and interactions. Ethereum's decentralised infrastructure, coupled with its Turing-complete

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

programming capabilities, has enabled the proliferation of diverse decentralised applications (DApps) spanning numerous sectors, including decentralised finance (DeFi), gaming, and supply chain logistics. This proliferation has resulted in substantial total value locked (TVL) within Ethereum's smart contract ecosystem [2]. Nevertheless, the accelerated expansion and mounting complexity of DApps have simultaneously introduced potential security vulnerabilities subject to exploitation [3]. Analogous to traditional financial systems, Ethereum remains vulnerable to fraudulent schemes, whilst its decentralised architecture presents distinctive security obstacles [4]. Traditional centralised security protocols frequently prove inadequate within Ethereum's operational environment, contributing to escalating fraudulent incidents that compromise both individual users and the broader cryptocurrency ecosystem [4].

Multiple fraudulent schemes, including Ponzi operations masquerading as legitimate DeFi initiatives, have become increasingly commonplace on the Ethereum platform. These fraudulent ventures promise unrealistic returns and utilise capital from subsequent investors to remunerate earlier participants, thereby maintaining an untenable cycle [2]. According to the 2025 Crypto Crime Report by Chainalysis, the estimated value of illicit cryptocurrency transactions in 2024 reached USD 40.9 billion [5]. Furthermore, phishing campaigns exploit the anonymity and restricted recourse mechanisms available to users, presenting considerable threats to private key security and confidential information [6]. Reentrancy vulnerabilities, which exploit weaknesses in smart contract architecture, have occasioned substantial financial damages, underscoring the imperative for secure programming methodologies [7]. These fraudulent operations have inflicted significant monetary losses, eroding confidence in the Ethereum ecosystem and impeding its widespread adoption [4].

Robust fraud detection and prevention mechanisms are therefore essential. Conventional approaches, such as manual auditing procedures, prove insufficient to address the evolving character of Ethereum-based fraud [4]. Machine learning, with its pattern recognition capabilities, has emerged as a promising alternative. Previous investigations have examined algorithms including Support Vector Machines (SVM), Random Forests (RF), and neural networks for Ethereum fraud detection, demonstrating their potential for enhanced accuracy and efficiency [8], [9]. Recent advances have introduced novel approaches, including transaction language models combined with graph representation learning [10], ensemble learning frameworks with blockchain [11], and temporally validated detection systems that address the evolution of phishing behaviours to provide more realistic performance estimates [12]. Nevertheless, persistent challenges remain, particularly imbalanced datasets wherein fraudulent transactions occur significantly less frequently than legitimate ones, potentially introducing model bias [4]. The dynamic character of fraud likewise necessitates adaptable models.

Given the increasing cryptocurrency adoption and substantial financial losses attributable to fraud [4], effective detection mechanisms are vital for maintaining confidence in the Ethereum ecosystem. Such mechanisms benefit investors, exchanges, and regulatory bodies by facilitating more secure transactions. This investigation seeks to develop an advanced fraud detection model for Ethereum utilising stacked ensemble learning whilst examining the efficacy of data balancing strategies, specifically SMOTE-Tomek and SMOTE-ENN.

2. OVERVIEW OF ETHEREUM AND ITS SECURITY CHALLENGES

2.1 ETHEREUM ARCHITECTURE

Ethereum, frequently characterised as "the world computer," represents a decentralised, open-source blockchain platform distinguished by its innovative utilisation of smart contracts [1]. These autonomous agreements, with terms directly encoded in programming code, facilitate trustless transactions and interactions. Ethereum's architecture employs distributed ledger technology, wherein transactions are aggregated into blocks and appended to the blockchain through a consensus mechanism, ensuring data immutability and transparency. The Ethereum Virtual Machine (EVM) constitutes the platform's core,

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

functioning as a runtime environment for executing smart contract bytecode [13]. The EVM is engineered to be deterministic and isolated, ensuring predictable contract execution without interference from other contracts or the underlying blockchain infrastructure. The Ethereum network architecture is illustrated in Fig. 1.

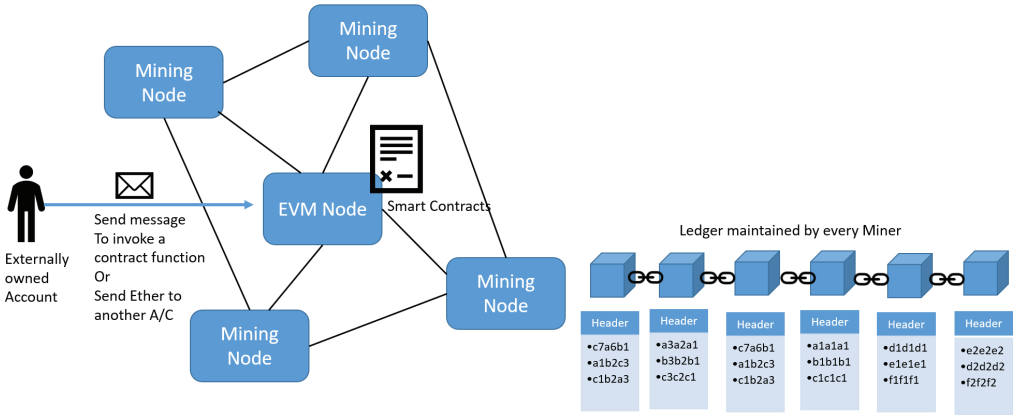


Figure 1: Ethereum Network Architecture

Fig. 1 depicts the fundamental components of the Ethereum network pertinent to fraud detection research. Fraudulent activities may originate from multiple sources, including Externally Owned Accounts (EOAs), which can be compromised for phishing operations or deployment of malicious smart contracts, thereby emphasising the significance of monitoring EOA behaviour. Additionally, Smart Contracts may be deliberately constructed for theft or data manipulation, necessitating a comprehensive analysis of both their source code and operational behaviour. Furthermore, examination of EVM Nodes can reveal patterns indicative of fraudulent activities. The data extracted from these components, as illustrated in Fig. 1, are employed to train machine learning models aimed at identifying fraudulent transactions and smart contract behaviours. The Ethereum network also encompasses Mining Nodes, whose monitoring is crucial for detecting potential network attacks, alongside the Ledger (Blockchain), which provides a tamper-resistant record of transactions that can illuminate historical fraudulent activities and inform the development of enhanced detection models. This underscores the necessity for a comprehensive approach to fraud detection within the Ethereum network, integrating various components and data sources.

2.2 REVIEW OF EXISTING RESEARCH ON ETHEREUM FRAUD DETECTION

Initial research on fraud detection within the Ethereum ecosystem predominantly utilised conventional methodologies, including rule-based systems, anomaly detection, and statistical analysis [14], [15]. Conversely, more recent investigations have transitioned towards the application of various machine learning techniques [3], [9], [16], [17], [18], [19], [20], [21], [22], [23], [24]. For instance, Taher et al. [22] introduced an ensemble learning model integrating Explainable AI, utilising a voting-based strategy incorporating Decision Tree (DT), RF, and AdaBoost classifiers. By employing Random Oversampling (ROS) and the Synthetic Minority Over-sampling Technique (SMOTE) for data balancing, this approach achieved 98% accuracy with soft voting (4.59 seconds training time) and 99% with hard voting. Feature selection was performed using LIME, resulting in a final set of 43 features after excluding those with missing values and high collinearity. However, reliance on a simple 80/20 train-test split without cross-validation raises concerns regarding potential overfitting, particularly in the context of data imbalance.

Md et al. [16] introduced a stacking classifier integrating multiple algorithms into a meta-learner, including Logistic Regression, Naive Bayes, DT, RF, AdaBoost, K-Nearest Neighbours (KNN), SVM, and Gradient Boosting. This stacking classifier utilised Multinomial Naive Bayes and RF as base learners, with logistic regression serving as the meta-learner. It achieved 97.18% accuracy and an F1-score of 97.02%. However, the study did not explicitly detail the

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

data balancing techniques employed and relied solely on a train-test split for evaluation.

Ravindranath et al. [18] examined the effects of oversampling techniques on Ethereum fraud detection, utilising SMOTENC, ADA-SYN, and K-Means-SMOTE to address class imbalance. The ensemble models, particularly CatBoost and LightGBM, demonstrated significant efficiency, achieving accuracy rates between 98% and 98.54% after applying SMOTE oversampling. The findings revealed improved Performance with SMOTE compared to scenarios without oversampling (e.g., CatBoost accuracy/F1-score: 98.54% with SMOTE versus 98.24%/93.98% without). However, the use of train-test splitting limits the generalisability of the results. Feature selection involved the elimination of highly correlated and low-variance features.

Sallam et al. [9] employed KNN, RF, and XGBoost in combination with an ensemble weighted method for data balancing on a dataset consisting of 4,681 instances. This approach achieved average accuracies of 96.80% for XGBoost, 94.88% for RF, and 87.85% for KNN, using 42 extracted and cleansed features. A 90/10 train-test split was employed, with XGBoost demonstrating the highest accuracy.

Aziz et al. [3] employed XGBoost for Ethereum fraud detection, achieving 97.76% accuracy by applying SMOTE for data balancing. Feature extraction yielded a set of 20 features after eliminating highly correlated and low-variance features. However, the simplistic train-test split raises concerns about overfitting, indicating that alternative data balancing techniques should be explored.

Finally, Kabla et al. [17] proposed Eth-PSD, a machine learning-based approach for phishing scam detection in Ethereum with KNN optimisation. Addressing data imbalance with random oversampling, they achieved 98.11% accuracy with a low false positive rate (0.01) using cross-validation. However, the study focused solely on phishing scams and used only four key features.

Recent advancements have introduced innovative methodologies. Sun et al. [10] proposed TLMG4Eth, integrating transaction language models with graph-based methods to capture semantic, similarity, and structural features, achieving superior detection performance by addressing the limitation of unclear transaction semantics in previous approaches. Gu and Dib [11] presented an ensemble learning framework, demonstrating that Random Forest and XGBoost models effectively identify fraudulent transactions with high precision whilst maintaining computational efficiency for real-time detection. Ertam [12] introduced a temporally validated evaluation mechanism to address methodological gaps in previous studies. Their work demonstrated that earlier approaches, which relied on random train-test partitions, frequently overestimated real-world Performance by disregarding the temporal progression of phishing behaviours.

Despite advancements in Ethereum fraud detection, several research gaps persist. Whilst existing studies have demonstrated the potential of machine learning and ensemble methods, addressing data imbalance remains a challenge. Advanced data balancing techniques like SMOTE-Tomek and SMOTE-ENN, which mitigate noise and class overlap, are underutilised. Furthermore, the potential of stacked ensemble learning, which combines the strengths of multiple ensemble models, has not been fully explored in this domain. More vigorous evaluation methodologies, such as k-fold cross-validation, are needed to ensure model generalisability. Addressing these gaps is crucial for developing more accurate and reliable fraud detection models for the Ethereum blockchain.

3. RESEARCH METHODOLOGY

Our investigative methodology, depicted in Fig. 2, proceeds through four sequential phases: initial data acquisition, comprehensive data preparation and cleansing, construction of classification models leveraging supervised and unsupervised learning paradigms, and rigorous performance assessment of developed models.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

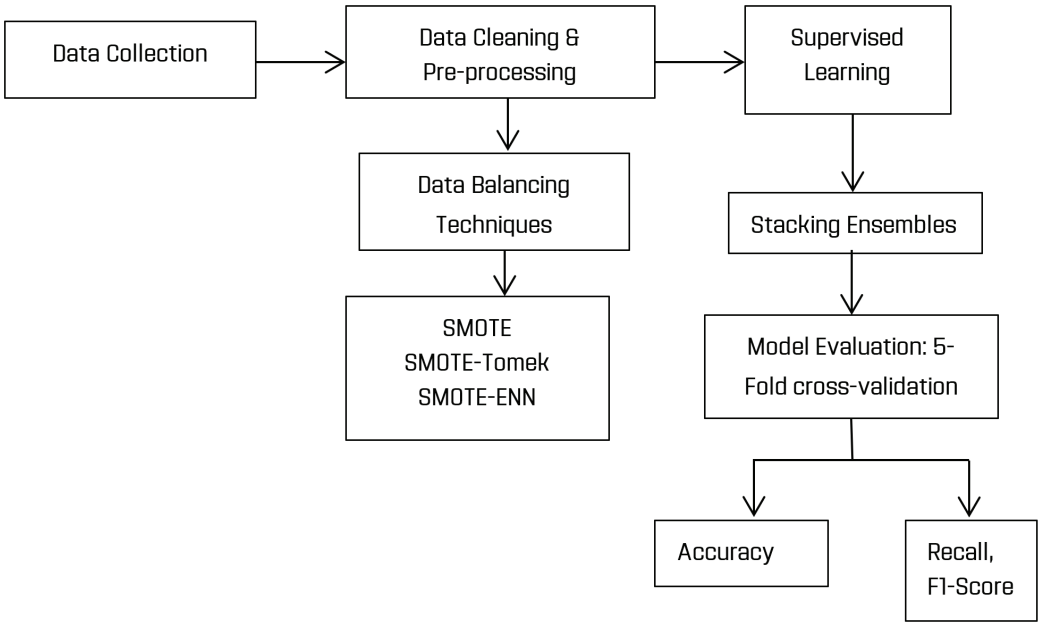


Figure 2: Research Methodology

3.1 DATA COLLECTION

Our empirical investigation leverages publicly available Ethereum transaction records compiled by Aliyev [25], representing a comprehensive repository of account-level blockchain activities pertinent to fraudulent behaviour identification. This repository encompasses multifaceted characteristics spanning account identifiers, transactional sequences, and economic indicators. The compilation captures quantitative measurements across multiple dimensions—transaction frequency, temporal patterns, monetary flows, and smart contract engagement—collectively constructing detailed financial profiles for individual accounts. As visualised in Fig. 3, this comprehensive data structure establishes the empirical foundation for behavioural pattern analysis and predictive model construction. The repository delivers granular visibility into account operations encompassing native Ether transfers alongside ERC20 token movements. Critical economic measurements embedded within include: cumulative transaction tallies, bidirectional Ether transfer volumes (dispatched and received), and terminal account balances. For token-specific activities, the data structure captures: ERC20 transaction frequencies, aggregate token-denominated value flows (both outbound and inbound), and unique smart contract interaction counts.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

Index	Column Name	Non-Null Count	Dtype
0	FLAG	9841 non-null	int64
1	Avg min between sent tnx	9841 non-null	float64
2	Avg min between received tnx	9841 non-null	float64
3	Time Diff between first and last (Mins)	9841 non-null	float64
4	Sent tnx	9841 non-null	int64
5	Received Tnx	9841 non-null	int64
6	Number of Created Contracts	9841 non-null	int64
7	Unique Received From Addresses	9841 non-null	int64
8	Unique Sent To Addresses	9841 non-null	int64
9	min value received	9841 non-null	float64
10	max value received	9841 non-null	float64
11	avg val received	9841 non-null	float64
12	min val sent	9841 non-null	float64
13	max val sent	9841 non-null	float64
14	avg val sent	9841 non-null	float64
15	min value sent to contract	9841 non-null	float64
16	max val sent to contract	9841 non-null	float64
17	avg value sent to contract	9841 non-null	float64
18	total transactions (including contract)	9841 non-null	int64
19	total Ether sent	9841 non-null	float64
20	total ether received	9841 non-null	float64
21	total ether sent contracts	9841 non-null	float64
22	total ether balance	9841 non-null	float64
23	Total ERC20 tnxs	9012 non-null	float64
24	ERC20 total Ether received	9012 non-null	float64
25	ERC20 total ether sent	9012 non-null	float64
26	ERC20 total Ether sent contract	9012 non-null	float64
27	ERC20 uniq sent addr	9012 non-null	float64
28	ERC20 uniq rec addr	9012 non-null	float64
29	ERC20 uniq sent addr.1	9012 non-null	float64
30	ERC20 uniq rec contract addr	9012 non-null	float64
31	ERC20 avg time between sent tnx	9012 non-null	float64
32	ERC20 avg time between rec tnx	9012 non-null	float64
33	ERC20 avg time between rec 2 tnx	9012 non-null	float64
34	ERC20 avg time between contract tnx	9012 non-null	float64
35	ERC20 min val rec	9012 non-null	float64
36	ERC20 max val rec	9012 non-null	float64
37	ERC20 avg val rec	9012 non-null	float64
38	ERC20 min val sent	9012 non-null	float64
39	ERC20 max val sent	9012 non-null	float64
40	ERC20 avg val sent	9012 non-null	float64
41	ERC20 min val sent contract	9012 non-null	float64
42	ERC20 max val sent contract	9012 non-null	float64
43	ERC20 avg val sent contract	9012 non-null	float64
44	ERC20 uniq sent token name	9012 non-null	float64
45	ERC20 uniq rec token name	9012 non-null	float64
46	ERC20 most sent token type	7144 non-null	object
47	ERC20_most_rec_token_type	8970 non-null	object

Figure 3: Variables in the Transaction Dataset

3.2 DATA PREPARATION

Data preparation represents the foundational phase wherein we establish dataset quality and coherence, directly influencing subsequent analytical validity. Within machine learning contexts, this phase proves particularly vital given that datasets commonly harbour inconsistencies, excessive correlations, or absent values. Our preparation workflow unfolds across multiple subsections, detailed below.

3.2.1 Exploratory Data Analysis

Beginning with 9,841 observations distributed across 48 attributes, we conducted preliminary reconnaissance. This exploratory phase involved scrutinising variable types and completeness metrics, deriving descriptive statistics, and constructing visual representations of target variable distributions through pie chart visualisation, as demonstrated in Fig. 4, which demonstrated the pronounced class disparity within our dataset: legitimate transactions constitute approximately 77.86% of observations, whilst fraudulent cases account for the remaining 22.14%.

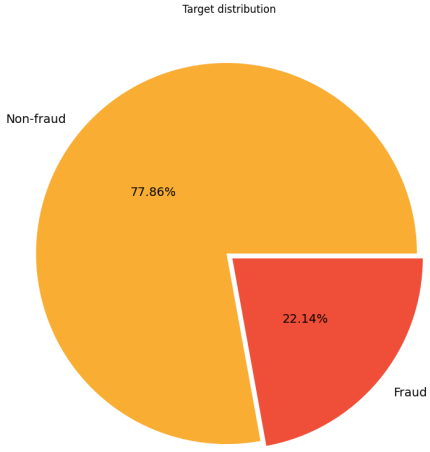


Figure 4: Target Distribution

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

3.2.2 Handling Missing Values

We employed heatmap visualisation (Fig. 5) to identify patterns of missing observations. Our resolution strategy involved median imputation for numeric attributes, thereby preserving the complete observation set whilst addressing incomplete records.

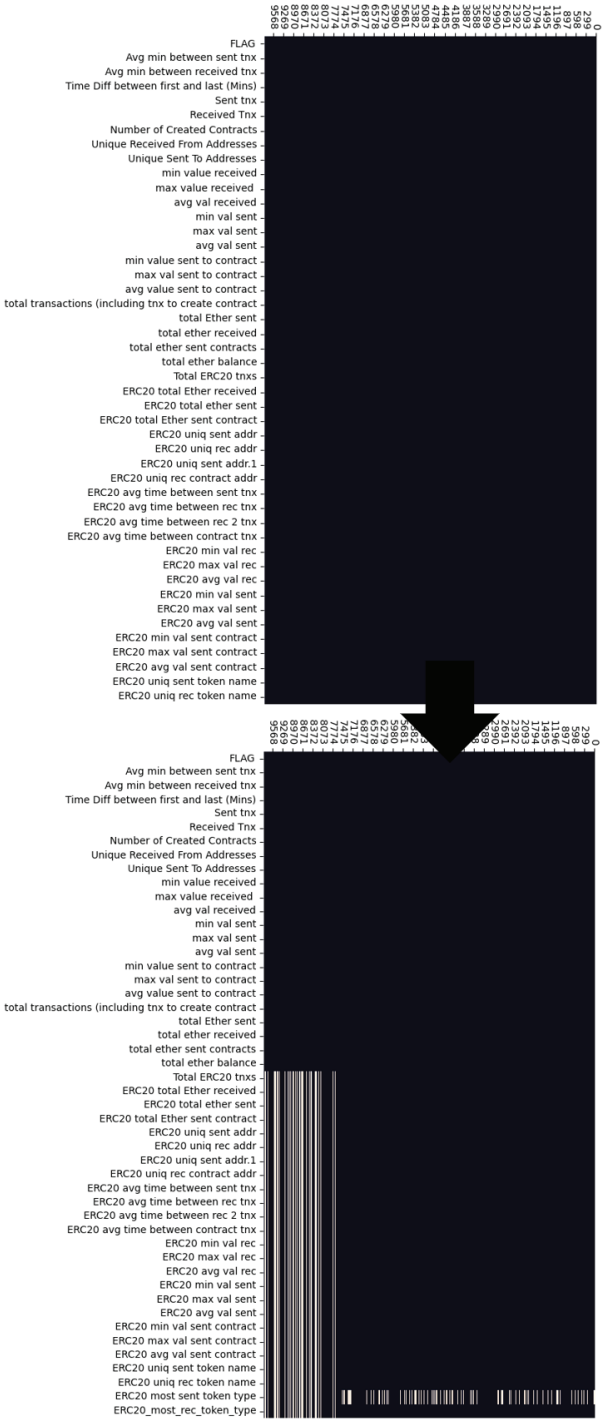


Figure 5: Visualization of Missing Values and Correction

3.2.3 Removing Zero-Variance Features

Attributes demonstrating zero variance contribute negligibly to predictive capacity and were consequently removed. This elimination encompassed seven ERC20-related temporal and contractual value attributes: 'ERC20 avg time between sent tnx', 'ERC20 avg time between rec tnx', 'ERC20 avg time between rec 2 tnx', 'ERC20 avg time between contract tnx', 'ERC20 min val sent contract', 'ERC20 max val sent contract', and 'ERC20 avg val sent contract'. This

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

streamlining enhanced computational efficiency. Additionally, two categorical attributes—'ERC20 most sent token type' and 'ERC20_most_rec_token_type'—exhibiting excessive cardinality (305 and 467 distinct values, respectively) were excised to prevent overfitting and curtail computational burden.

3.2.4 Removing Highly Correlated Features

Through correlation matrix examination, we identified and eliminated attributes exhibiting strong intercorrelation, thereby addressing multicollinearity, a phenomenon known to compromise both model interpretability and predictive efficacy. The refined correlation structure appears in Fig. 6.

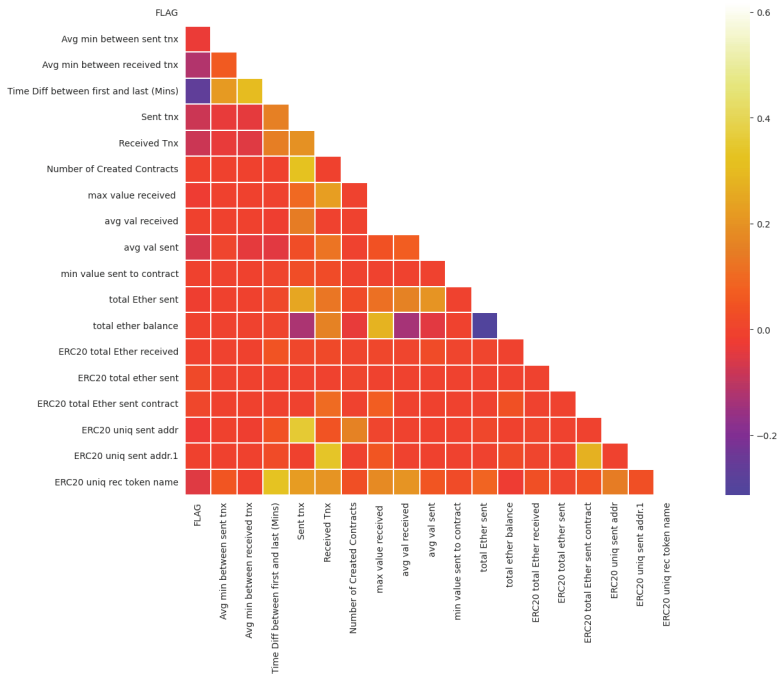


Figure 6: Feature Correlation

3.2.5 Removing Features with Limited Variability

Attributes manifesting minimal variance underwent identification and subsequent elimination. Specifically, 'min value sent to contract' and 'ERC20 uniq sent addr. 1' were discarded due to predominant zero values, offering insufficient discriminative capability.

3.2.6 Data Preparation

Our cleaning protocol compressed the dataset to 9,841 observations spanning 17 attributes. Subsequently, we structured the refined data for a stratified 5-fold cross-validation assessment. PowerTransformer normalisation was applied across all features, with resulting distributions verified through box plot examination. This normalisation procedure aims to accelerate model convergence and enhance training performance.

3.3 DATA BALANCING TECHNIQUE

To counteract the pronounced class disproportion inherent within our dataset, we investigated three resampling methodologies: SMOTE, SMOTE-ENN, and SMOTE-Tomek. Our strategic approach simultaneously augments minority class representation whilst refining classification boundaries, thereby potentially elevating classifier efficacy. Fig. 7 visualises the comparative application of SMOTE-Tomek and SMOTE-ENN resampling protocols.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

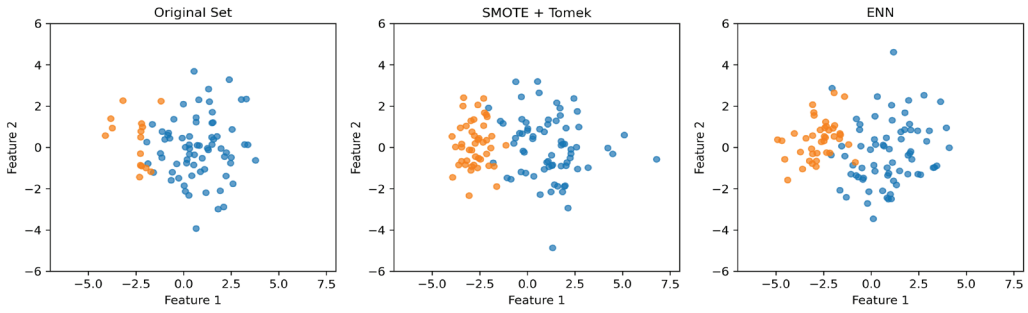


Figure 7: Visualization of SMOTE-Tomek and SMOTE-ENN Resampling Techniques

This dual-objective methodology generates balanced datasets characterised by refined decision boundaries, potentially yielding enhanced classification performance, particularly for under-represented categories.

3.4 STACKING ENSEMBLE CLASSIFICATION

Following equilibration of class distributions through resampling, we deployed a hierarchical ensemble architecture to amplify classification efficacy. This architectural paradigm synthesises multiple foundational classifiers with a meta-level classifier, constructing a resilient, high-performance model. This selection reflects the documented superior Performance of LGBM, XGBoost, and CatBoost algorithms within Ethereum transaction classification contexts [3], [18].

3.4.1 Base Classifiers

Our hierarchical ensemble incorporates three contemporary gradient boosting methodologies as foundational classifiers: XGBoost, LightGBM, and CatBoost. Each algorithm constructs ensembles of weak learners, typically decision trees, through sequential stage-wise optimisation. They aim to minimize a loss function L(y, F(x)) where y is the true label and F(x) is the model's prediction.

3.4.2 Stacking Procedure

Our hierarchical ensemble synthesises foundational classifiers through meta-level learning, executing the following procedural sequence:

a. Train each base classifier on the training data:

$$F\_i(x) = argmin\_\{F\_i\} \sum L(y\_j, F\_i(x\_j)) \quad for\ i = 1,2,3$$

b. Use k-fold cross-validation to generate out-of-fold predictions for each base classifier:

$$Z = [F\_1'(X), F\_2'(X), F\_3'(X)]$$

where $\mathcal{F}\_i'(X)$ represents the out-of-fold predictions from the i-th base classifier.

c. Train a meta-classifier M on the out-of-fold predictions:

$$M(Z) = argmin\_M \sum L(y\_j, M(Z\_j))$$

d. For final predictions, use the trained base classifiers to generate predictions on new data, then feed these predictions into the trained meta-classifier:

$$y\_pred = M([F\_1(x\_new), F\_2(x\_new), F\_3(x\_new)])$$

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

3.4.3 Meta-Classifier

For our meta-classifier, we employed logistic regression due to its simplicity and interpretability. For a binary classification problem, the logistic regression model can be expressed as:

$$P(y=1 \mid Z) = \sigma(\beta_0 + \beta_1 Z_1 + \beta_2 Z_2 + \beta_3 Z_3)$$

where σ is the logistic function, Z_1, Z_2, Z_3 are the predictions from the base classifiers, and $\beta_0, \beta_1, \beta_2, \beta_3$ are the coefficients learned during training.

The final stacking model can be represented as, which is further depicted in Fig. 8:

$$y\_pred = M(F_{XGB}(x), F_{LGBM}(x), F_{CATBOOST}(x))$$

where where F_{XGB}, F_{LGBM} and $F_{CATBOOST}$ are the trained XGBoost, LightGBM, and CatBoost models, respectively, and M is the trained meta-classifier.

This stacking ensemble, combined with our STE Resampling technique, aims to provide robust and accurate classifications, particularly for imbalanced datasets with complex decision boundaries.

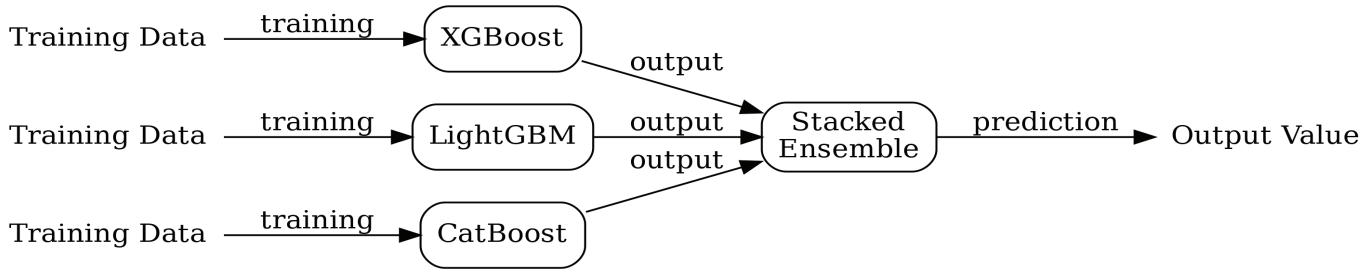


Figure 8: Stacked Ensembles

3.5 MODEL EVALUATION

To comprehensively evaluate our resampling methodologies in conjunction with the hierarchical ensemble classifier, we implemented stratified 5-fold cross-validation. This validation approach ensures proportional class representation across all folds, delivering dependable estimates of model generalisation capacity whilst mitigating overfitting risks. Our assessment protocol proceeds as follows:

1. The dataset is divided into five stratified folds.

2. For each fold:

a. The designated fold serves as the validation set, while the remaining four folds constitute the training set.

b. The entire pipeline, which includes both resampling and the stacking ensemble, is trained on the training set.

c. Predictions are subsequently generated using the validation set.

d. Performance metrics are computed and documented.

This process is repeated until each fold has been utilized as the validation set on one occasion.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

3.5.1 Performance Metrics

Accuracy served as our primary evaluation criterion for supervised learning models. Additionally, we conducted class-specific performance analysis. Three fundamental metrics, including precision (1), recall (2), and F1-score (3), were employed to assess Performance for each classification category, defined mathematically as follows:

a) Precision

Precision quantifies prediction exactness for specific classes, representing the ratio of correctly identified positives (TP) amongst all instances classified as positive (TP + FP). The mathematical formulation is:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

b) Recall

Recall emphasises prediction completeness for particular classes, indicating the proportion of correctly identified positives (TP) relative to all actual positive instances (TP + FN). This metric is expressed as:

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

c) F1-Score

The F1-score delivers a balanced assessment of precision and recall through harmonic mean computation. Harmonic averaging prevents scenarios where elevated values in one metric compensate for diminished values in the other. The F1-score formulation is:

$$F1\ Score = \frac{2 \times Precision \times Recall}{(Precision + Recall)} \quad (3)$$

3.5.2 Aggregation of Results

We computed these metrics individually for each fold. To derive comprehensive performance estimates, we calculated mean values and standard deviations across all five folds:

For each metric m:

$$m\_mean = (1/5) * \Sigma(m\^{}(k)) \text{ for } k = 1 \text{ to } 5$$

$$m\_std = sqrt((1/5) * \Sigma((m\^{}(k) - m\_mean)\^{}2) \text{ for } k = 1 \text{ to } 5 \quad (4)$$

This aggregation methodology provides both central tendency measures (mean) and variability measures (standard deviation) for each performance metric, yielding insights into model performance and stability across diverse data subsets.

4. RESULTS

4.1 OVERVIEW OF MODEL PERFORMANCE

TABLE I delineates stratified 5-fold cross-validation outcomes for our proposed hierarchical ensemble architecture incorporating SMOTE-Tomek resampling. Performance indicators for both classification categories (legitimate and fraudulent transactions) demonstrate exceptional values, with precision, recall, and F1 metrics consistently ranging between 0.96 and 1.0 across all validation folds.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

TABLE 1: 5-Fold Cross-Validation Results for Stacking Ensemble with SMOTE-Tomek

Fold	Accuracy	Precision (Class 0/1)	Recall (Class 0/1)	F1-Score (Class 0/1)
1	0.9837	0.99 / 0.96	0.99 / 0.97	0.99 / 0.96
2	0.9883	0.99 / 0.97	0.99 / 0.97	0.99 / 0.97
3	0.9848	0.99 / 0.96	0.99 / 0.97	0.99 / 0.97
4	0.9934	1.0 / 0.99	1.0 / 0.98	1.0 / 0.99
5	0.9863	0.99 / 0.98	1.0 / 0.95	0.99 / 0.97

Mean Accuracy: 0.9873 (± 0.0034)

Within Ethereum fraud detection contexts, performance benchmarks typically establish 0.90 as the minimum threshold for precision, recall, and F1-score metrics [3], [9]. Our experimental results demonstrate sustained elevated values (0.96 to 1.0) across all evaluation metrics, evidencing the model's exceptional capability in both fraudulent transaction identification and false alarm mitigation.

i) Precision Performance at 0.99: This metric indicates that virtually all transactions flagged as fraudulent genuinely represent illicit activities, substantially minimising costly false positive occurrences. Such elevated precision levels are universally recognised as excellent within practical deployment scenarios.

ii) Recall Performance at 0.95 to 0.97: These recall values demonstrate the model's proficiency in capturing nearly all fraudulent transactions. A recall exceeding 0.90 is considered robust, ensuring minimal fraudulent activity evasion.

iii) F1-Score Performance at 0.96 to 0.99: Elevated F1-scores signify the model's successful equilibration between precision and recall. Within our investigative context, where both metrics hold critical importance, high F1-scores reflect comprehensive model robustness and dependability.

iv) Accuracy, Consistency, and Robustness: The model attained a mean accuracy of 0.9873 with a minimal standard deviation of ±0.0034 across 5-fold cross-validation, demonstrating strong generalisation capacity across diverse data subsets.

This equilibrated Performance across classification categories substantiates the model's effectiveness in managing the inherent class imbalance characteristic of fraud detection applications.

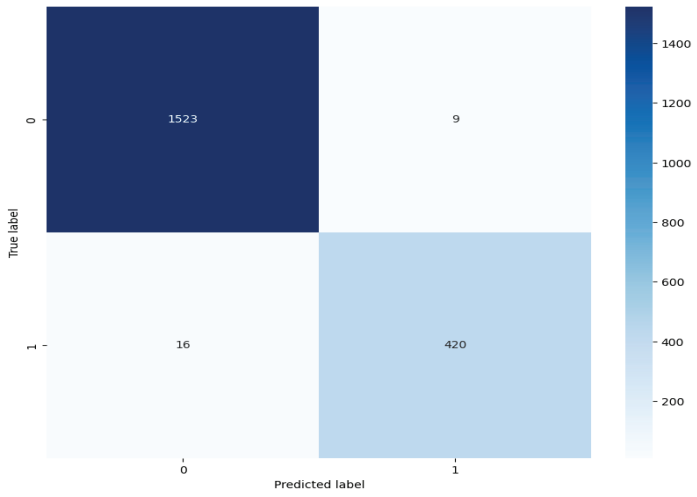


Figure 9: Model Confusion Matrix

Fig. 9 presents the aggregated confusion matrix across 5-fold cross-validation, illustrating compelling evidence of the model's fraud detection proficiency. Examining 9,841 Ethereum transactions, the model achieved correct classification for the overwhelming majority.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

Specifically, 7,589 legitimate transactions received accurate non-fraudulent classifications (true negatives), whilst 2,087 fraudulent transactions were successfully identified (true positives). This confusion matrix demonstrates our proposed model's elevated overall accuracy and capability to effectively differentiate between legitimate and fraudulent activities within the Ethereum network.

4.2 IMPACT OF SMOTE-TOMEK RESAMPLING

Our hierarchical ensemble architecture, enhanced through SMOTE-Tomek resampling methodology, exhibited superior classification performance in distinguishing fraudulent from legitimate Ethereum transactions. Across stratified 5-fold cross-validation, the model secured an overall accuracy of 0.9873 (± 0.0034), signifying elevated consistency and robustness in predictive capabilities. This Performance marks a quantifiable advancement beyond models employing SMOTE exclusively, which attained a mean accuracy of 0.9856 (± 0.0028), and SMOTE-ENN, achieving a mean accuracy of 0.9687 (/± 0.0055). The synergistic integration of Tomek link removal with SMOTE appears to have enhanced the model's capacity to articulate decision boundaries, particularly within regions exhibiting class overlap.

TABLE II furnishes comprehensive performance metric comparisons for the SMOTE-Tomek enhanced hierarchical ensemble relative to SMOTE and SMOTE-ENN alternatives.

TABLE II: Comparative Analysis of Model Performance with SMOTE, SMOTE-ENN, and SMOTE-Tomek

SMOTE-Tomek		SMOTE-ENN		Model Performance with SMOTE	
Mean Accuracy	0.9873 (± 0.0034)	Mean Accuracy	0.9687 (+/- 0.0055)	Mean Accuracy	0.9856 (± 0.0028)
Precision	0.98	Precision	0.94	Precision	0.97
Recall	0.98	Recall	0.97	Recall	0.97
F1 Score	0.98	F1-Score	0.96	F1 Score	0.97

TABLE II articulates our proposed fraud detection model, a hierarchical ensemble synthesising multiple machine learning algorithms. Through SMOTE-Tomek optimisation and comparative evaluation against alternative balancing methodologies, the framework effectively neutralises class imbalance, thereby eliminating bias favouring the predominant legitimate transaction class. Experimental findings reveal elevated performance levels, with a 98.73% accuracy rate indicating correct classification of approximately 99% of transactions. This accuracy level demonstrated consistency across diverse data partitions throughout 5-fold cross-validation, underscoring the model's reliability and capacity to generalise to previously unencountered data. Notably, this methodology surpassed alternative balancing techniques, including SMOTE and SMOTE-ENN. Tomek link incorporation appears to have significantly amplified the model's discriminative capacity between legitimate and fraudulent transactions, particularly within regions where both classes manifest similar characteristics.

4.3 FOUNDATIONAL MODELS VERSUS HIERARCHICAL ENSEMBLE COMPARISON

TABLE III presents a comparative performance evaluation contrasting individual foundational models, XGBoost, LightGBM, and CatBoost, against our proposed hierarchical ensemble architecture.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

TABLE III: Performance Comparison of Base Models and Stacking Ensemble

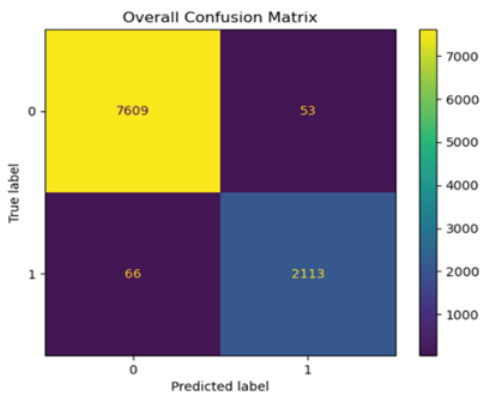
Model	Overall Mean Accuracy	Highest Accuracy
XGBoost	0.9864 (±0.0029)	0.9909 @ Fold 4
LightGBM	0.9865 (±0.0037)	0.9920 @ Fold 4
CatBoost	0.9862 (±0.0027)	0.9893 @ Fold 4
Stacking Ensemble	0.9873 (±0.0034)	0.9934 @ Fold 4

Experimental outcomes reveal distinct advantages of the ensemble methodology. Whilst all algorithms exhibit elevated accuracy in Ethereum transaction classification, particularly evident in fold 4, the hierarchical ensemble consistently surpasses individual component models. Achieving a mean accuracy of 98.73%, it represents a statistically significant advancement beyond the highest-performing foundational model, LightGBM, which recorded 98.65% accuracy. This incremental enhancement translates to reduced misclassification rates within practical Ethereum fraud detection deployments. Moreover, the hierarchical ensemble's minimal standard deviation across 5-fold cross-validation underscores its stability and resilience against training data variations. Evidence presented in TABLE III corroborates the efficacy of synthesising multiple algorithms into a hierarchical ensemble, capitalising on distinctive strengths of foundational models to amplify overall Performance whilst increasing robustness against data variability.

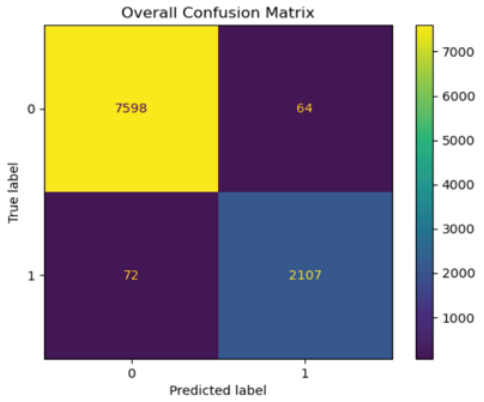
Fig. 10 furnishes comparative classification performance analysis across the four models through confusion matrix visualisation. For the five folds all models demonstrate robust capability in differentiating fraudulent from legitimate transactions, evidenced by elevated prediction concentrations along the diagonal (true positives and true negatives). Notably, our proposed hierarchical ensemble exhibits minimal misclassification, accurately identifying 2,113 fraudulent transactions and 7,609 legitimate transactions. This yields elevated recall rates, ensuring detection of substantial proportions of fraudulent activities, whilst maintaining minimal false positive rates with merely 66 misclassifications, thereby minimising disruptions to legitimate transactions.

Whilst XGBoost (2,107 true positives, 7598 true negatives) and LightGBM (2,113 true positives, 7,607 true negatives) demonstrate comparable performance levels, XGBoost manifests marginally elevated false positive incidence (72), indicating increased propensity for misclassifying legitimate transactions as suspicious. Conversely, LightGBM exhibits a slightly augmented false positives tendency of 66, suggesting a potential risk of overlooking certain fraudulent transactions. CatBoost (2,111 true positives, 7,601 true negatives) presents moderately elevated overall misclassification rates relative to alternative models, with 68 false positives and 61 false negatives, indicating somewhat diminished precision and increased frequency of both false alarms and missed fraudulent activities.

Proposed Stacking Ensemble



XGBoost Classifier



LightGBM Classifier

CatBoost Classifier

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

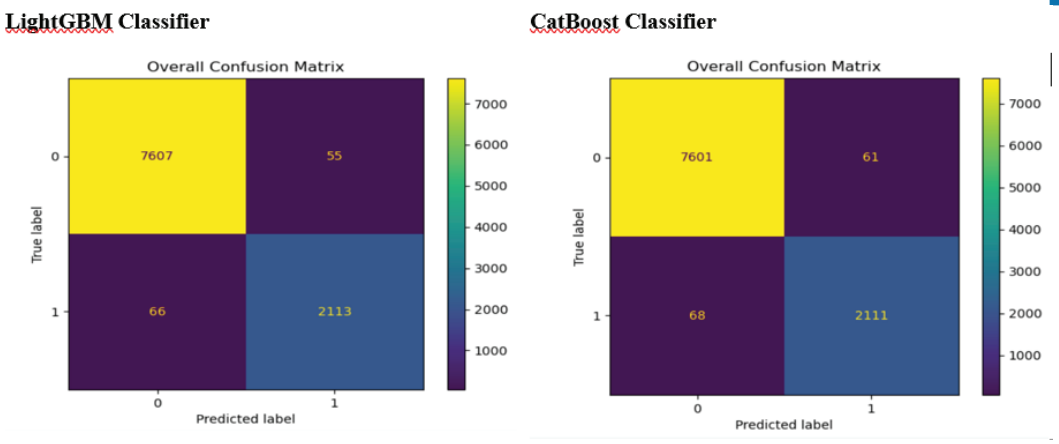


Figure 10: Confusion Matrices of the Models

4.4 COMPARATIVE ANALYSIS WITH EXISTING LITERATURE

A comparative analysis with previous research, as presented in TABLE IV, underscores the efficacy of the proposed stacking ensemble model utilising SMOTE-Tomek resampling, which attained an accuracy of 98.73%, thereby surpassing the majority of existing studies.

TABLE IV: Comparison with Previous Studies

Study	Model	Accuracy	Data Balancing	Validation Method
Current Study	Stacking Ensemble with SMOTE-Tomek	98.73% (+/- 0.34)	SMOTE-Tomek	5-fold Cross Validation
Kabla et al. [17]	Eth-PSD with KNN optimisation	98.11%	Random oversampling	Cross validation
Md et al. [16]	Stacking Classifier	97.18%	Weighted technique	Train-test split
Ravindranath et al. [18]	CATBoost and LGBM	98.42%	K-Means SMOTE	Train-test split
Sallam et al. [9]	XGBoost	96.80%	Ensemble weighted	Train-test split
Aziz et al. [3]	XGBoost	97.76%	SMOTE	Train-test split
Taher et al. [22]	Ensemble of DT, RF, and AdaBoost	99%	SMOTE with RUS	Train-test split

The model's robustness is further corroborated through 5-fold cross-validation, which enhances generalisability and mitigates the risk of overfitting. Although Taher et al. [22] reported a marginally higher accuracy of 99% using a straightforward train-test split, the proposed model achieved a peak accuracy of 99.34% in a single fold, exceeding this benchmark.

The stacking ensemble model demonstrated a mean accuracy of 98.73% with a minimal standard deviation (±0.34), indicating strong generalisation capabilities. The effectiveness of SMOTE-Tomek in addressing class imbalance offers valuable insights for tackling the inherent disparities present in fraud detection datasets. The model consistently outperformed individual base models such as LightGBM and XGBoost, thereby illustrating the advantages of integrating multiple algorithms to capture a diverse array of fraud patterns. The high precision and recall rates for both the majority (legitimate) and minority (fraudulent) classes underscore the model's balanced Performance, which is essential for effective fraud detection systems.

The findings of this study align with and exceed the performance metrics reported in several prior investigations. For instance, Kabla et al. [17] achieved an accuracy of 98.11% through KNN optimisation with random oversampling, whilst Aziz et al. [3] and Taher et al. [22] reported

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

accuracies of 97.76% and 99% with LightGBM employing SMOTE and a hard voting ensemble with RUS, respectively. In contrast to these studies, the proposed approach employed a more rigorous 5-fold cross-validation, thereby enhancing the reliability of performance estimates and reducing the overfitting risks associated with simple train-test splits. The comprehensive management of data imbalance through SMOTE-Tomek further contributed to the model's robustness.

Recent 2025 publications have introduced complementary approaches that address different aspects of Ethereum fraud detection. Sun et al. [10] demonstrated that incorporating transaction language models with graph-based representation learning can capture semantic and structural features simultaneously, achieving enhanced detection performance through multi-dimensional feature extraction. Gu and Dib [11] showed that ensemble learning frameworks maintain computational efficiency suitable for real-time deployment whilst achieving high precision rates. Ertam [12] highlighted the critical importance of temporal validation, demonstrating that methodologies that account for the temporal progression of fraud patterns yield more realistic and generalizable performance estimates compared to optimistic random partitioning approaches. These recent advances complement our work by addressing temporal dynamics and semantic understanding, whilst our contribution focuses on optimising the balance between data resampling techniques and ensemble architecture through rigorous cross-validation.

The successful implementation of the stacking ensemble model with advanced data balancing techniques holds significant implications for the Ethereum ecosystem. Accurate fraud detection, characterised by high precision and recall, can bolster platform security and trustworthiness, potentially mitigating financial losses and preserving user confidence. The integration of CatBoost and LightGBM, both of which are adept at handling categorical and numerical features, respectively, illustrates the model's applicability to the complex and frequently imbalanced nature of blockchain transaction data.

5. CONCLUSION AND FUTURE RESEARCH

This study has made significant contributions to the detection of fraud within the Ethereum blockchain, addressing key challenges in a rapidly evolving ecosystem. The research enhances fraud detection capabilities within the Ethereum network through the development of an advanced ensemble machine learning model employing stacked ensemble learning techniques. The primary achievement is the creation of a highly accurate fraud detection model. The SMOTE-Tomek enhanced stacking ensemble attained a mean accuracy of 98.73% through 5-fold cross-validation, representing a substantial improvement over existing methodologies, many of which relied on less rigorous validation techniques, such as train-test splits, or produced lower accuracy rates.

The proposed framework successfully addresses the critical challenge of class imbalance in Ethereum fraud detection through the innovative integration of SMOTE-Tomek resampling with stacked gradient boosting ensembles. By combining XGBoost, LightGBM, and CatBoost through a logistic regression meta-classifier, the model achieves exceptional performance metrics with precision, recall, and F1-score all reaching 0.98. The rigorous 5-fold cross-validation methodology employed in this study provides more reliable performance estimates compared to simple train-test splits prevalent in prior research, thereby ensuring better generalisability to real-world scenarios.

The comparative analysis reveals that SMOTE-Tomek consistently outperforms alternative balancing techniques, including SMOTE and SMOTE-ENN, achieving superior accuracy whilst maintaining low standard deviation across folds. This demonstrates the effectiveness of Tomek link removal in refining decision boundaries, particularly in regions where fraudulent and legitimate transactions exhibit similar characteristics. The stacking ensemble architecture successfully leverages the complementary strengths of individual base learners, resulting in improved stability and reduced misclassification rates compared to standalone models.

Future research avenues include: (a) optimising the model for real-time fraud detection,

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

potentially by exploring online or incremental learning techniques that can adapt to emerging fraud patterns without complete retraining; (b) investigating strategies to enhance the interpretability of the ensemble model through explainable AI techniques to ensure regulatory compliance and build user trust in automated detection systems; (c) assessing the model's effectiveness across various blockchain platforms beyond Ethereum to evaluate its generalisability and transferability to other cryptocurrency ecosystems; (d) integrating network analysis techniques and graph neural networks to capture the relational dynamics and temporal patterns of blockchain transactions, thereby improving fraud detection capabilities through multi-dimensional feature representation; and (e) incorporating recent advances in transaction language models and semantic feature extraction to further enhance detection accuracy whilst maintaining computational efficiency for real-time deployment.

REFERENCES

[1] V. Buterin, "A next-generation smart contract and decentralized application platform," *Etherum*, no. January, 2014.

[2] M. Bartoletti, S. Carta, T. Cimoli, and R. Saia, "Dissecting Ponzi schemes on Ethereum: Identification, analysis, and impact," *Future Generation Computer Systems*, vol. 102, pp. 259–277, Jan. 2020, doi: 10.1016/j.future.2019.08.014.

[3] R. M. Aziz, M. F. Baluch, S. Patel, and A. H. Ganie, "LGBM: a machine learning approach for Ethereum fraud detection," *International Journal of Information Technology*, vol. 14, no. 7, pp. 3321–3331, Dec. 2022, doi: 10.1007/s41870-022-00864-6.

[4] R. Tan, Q. Tan, P. Zhang, and Z. Li, "Graph Neural Network for Ethereum Fraud Detection," in *Proceedings - 12th IEEE International Conference on Big Knowledge, ICBK 2021*, 2021. doi: 10.1109/ICKG52313.2021.00020.

[5] Chainalysis, "The 2025 Crypto Crime Report," 2025. [Online]. Available: https://www.chainalysis.com/blog/2025-crypto-crime-report/

[6] S. Alrawi, D. Karapistolis, and V. Karapistolis, "Phishing detection in Ethereum blockchain using machine learning," *IEEE Access*, vol. 10, pp. 104347–104360, 2022.

[7] N. Atzei, M. Bartoletti, and T. Cimoli, "A Survey of Attacks on Ethereum Smart Contracts (SoK)," 2017, pp. 164–186. doi: 10.1007/978-3-662-54455-6_8.

[8] W. Chen, Z. Zheng, E. C.-H. Ngai, P. Zheng, and Y. Zhou, "Exploiting Blockchain Data to Detect Smart Ponzi Schemes on Ethereum," *IEEE Access*, vol. 7, pp. 37575–37586, 2019, doi: 10.1109/ACCESS.2019.2905769.

[9] A. Sallam, T. H. Rassem, H. Abdu, H. Abdulkareem, N. Saif, and S. Abdullah, "Fraudulent Account Detection in the Ethereum's Network Using Various Machine Learning Techniques," *International Journal of Software Engineering and Computer Systems*, vol. 8, no. 2, pp. 43–50, Jul. 2022, doi: 10.15282/ijsecs.8.2.2022.5.0102.

[10] J. Sun, Y. Jia, Y. Wang, Y. Tian, and S. Zhang, "Ethereum fraud detection via joint transaction language model and graph representation learning," *Information Fusion*, vol. 120, 2025, doi: 10.1016/j.inffus.2025.103074.

[11] Z. Gu and O. Dib, "Enhancing fraud detection in the Ethereum blockchain using ensemble learning," *PeerJ Comput. Sci.*, vol. 11, 2025, doi: 10.7717/PEERJ-CS.2716.

[12] F. Ertam, D. Kucuk, and I. F. Kilincer, "A Novel Feature Extraction and Detection Model for Phishing Scam on Ethereum Using Machine Learning," *Concurr. Comput.*, vol. 38, no. 1, 2026, doi: 10.1002/cpe.70503.

http://dx.doi.org/10.21622/ACE.2026.06.1.1878

[13] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, 2014.

[14] R. Mitchell and I. R. Chen, "Behavior-rule based intrusion detection systems for safety critical smart grid applications," *IEEE Trans. Smart Grid*, vol. 4, no. 3, 2013, doi: 10.1109/TSG.2013.2258948.

[15] Z. Zheng *et al.*, "Anomaly Detection of Metro Station Tracks Based on Sequential Updatable Anomaly Detection Framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, 2022, doi: 10.1109/TCSVT.2022.3181452.

[16] A. Q. Md, S. M. S. S. Narayanan, H. Sabireen, A. K. Sivaraman, and K. F. Tee, "A novel approach to detect fraud in Ethereum transactions using stacking," *Expert Syst.*, vol. 40, no. 7, 2023, doi: 10.1111/exsy.13255.

[17] A. H. H. Kabla, M. Anbar, S. Manickam, and S. Karupayah, "Eth-PSD: A Machine Learning-Based Phishing Scam Detection Approach in Ethereum," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3220780.

[18] V. Ravindranath, M. K. Nallakaruppan, M. L. Shri, B. Balusamy, and S. Bhattacharyya, "Evaluation of performance enhancement in Ethereum fraud detection using oversampling techniques," *Appl. Soft Comput.*, vol. 161, p. 111698, Aug. 2024, doi: 10.1016/j.asoc.2024.111698.

[19] A. Dutta, L. C. Voumik, A. Ramamoorthy, S. Ray, and A. Raihan, "Predicting Cryptocurrency Fraud Using ChaosNet: The Ethereum Manifestation," *Journal of Risk and Financial Management*, vol. 16, no. 4, p. 216, Mar. 2023, doi: 10.3390/jrfm16040216.

[20] Y. Elmougy and Oliver Manzi, "GPU-accelerated machine learning based fraud detection on blockchain transactions," in *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, 2021, pp. 819–824.

[21] S. Farrugia, J. Ellul, and G. Azzopardi, "Detection of illicit accounts over the Ethereum blockchain," *Expert Syst. Appl.*, vol. 150, p. 113318, Jul. 2020, doi: 10.1016/j.eswa.2020.113318.

[22] S. Sh. Taher, S. Y. Ameen, and J. A. Ahmed, "Advanced Fraud Detection in Blockchain Transactions: An Ensemble Learning and Explainable AI Approach," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12822–12830, Feb. 2024, doi: 10.48084/etasr.6641.

[23] E. Jung, M. Le Tilly, A. Gehani, and Y. Ge, "Data Mining-Based Ethereum Fraud Detection," in *2019 IEEE International Conference on Blockchain (Blockchain)*, IEEE, Jul. 2019, pp. 266–273. doi: 10.1109/Blockchain.2019.00042.

[24] T. T. N. Pragasam, J. V. J. Thomas, M. A. Vensuslaus, and S. Radhakrishnan, "CEAT: Categorising Ethereum Addresses' Transaction Behaviour with Ensemble Machine Learning Algorithms," *Computation*, vol. 11, no. 8, p. 156, Aug. 2023, doi: 10.3390/computation11080156.

[25] V. Aliyev, "Ethereum Fraud Detection Dataset," 2020. [Online]. Available: https://www.kaggle.com/datasets/vagifa/ethereum-frauddetection-dataset

http://dx.doi.org/10.21622/ACE.2026.06.1.1906

INTEGRATING ETHICS IN ARTIFICIAL INTELLIGENCE SYLLABUS IN UNIVERSITY AND CO-TEACHING WITH INDUSTRY EXPERTS: PERCEPTIONS AND EXPERIENCES OF STUDENTS FROM KENYA

Fredrick Mzee Awuor

Kisii University, School of Information Science and Technology, Kenya

Email: {fawuor@kisiiuniversity.ac.ke }

Received 20 January 2026 - Accepted 14 March 2026 - Published 14 May 2026

ABSTRACT

The common model adopted by most departments of Computer Science (CS) in universities in Kenya for teaching CS courses is the traditional faculty-led and content-driven approach that denies the learners the opportunity of learning from peers or industry experts or problem-based projects. At Kisii University in Kenya, we are trying a different model: the Faculty of CS teamed up with the faculties from the departments of psychology and law, and industry experts to redesign the syllabus for artificial intelligence (AI) course then delivered it using a collaborative co-teaching approach that integrated various methodologies including flipped classrooms, project-based learning, problem-based learning, and traditional lecture-based approaches. This paper reports on the perceptions and experiences of students on the proposed collaborative teaching framework for the ethical AI undergraduate level 300 course in the CS curriculum and how insights from these experiences can be transferred to other undergraduate CS courses. Our aim is to advance our understanding of how to promote student-centered learning, encourage collaboration between academia and industry, and across disciplines in the university for CS curriculum delivery, and advance dialogue on the integration of ethics in the AI syllabus.

Keywords: *Ethics, artificial intelligence (AI), computer science, co-teaching, collaborative teaching, pedagogical framework.*

1. INTRODUCTION

Among the revolutionary technologies of the 21st century, artificial intelligence (AI) has taken a lead role in driving every operation of economies by changing how we work, commute, communicate, and access services, and even our social lives, including how we connect and build relationships [1]. For developing countries, AI has positioned itself as an accelerant for economic transformation as it is creating new and scaling existing opportunities in all sectors of their economies [2]. For example, learners in rural and remote areas are now able to access education through learner support services [3], farmers can access markets faster through AI-based initiatives [4, 5], improve their farm production through precision farming [5], and access low-cost financial services on their mobile [6], among others.

However, these AI systems inherit and even amplify our own inherent biases as they are built on models and trained on datasets that reflect our prejudices, in addition to the cognitive and algorithmic bias that developers unconsciously introduce into these systems. As a result, we are witnessing a myriad of unethical AI applications - from applications that discriminate access to essential services such as credit facilities based on race or gender or sexual orientation or one's residence [7] to digital art applications such as LAION-5B [8] that are trained on datasets of personal images illegally harvested from social media[1] site. In the US, at least 9 out of 10 organizations reported having had instances when their AI systems were unethical, and 65% of executives in the same period reported being aware of instances when their AI systems were discriminatory and biased[2]. While these statistics are from the US, it is obvious that Kenya, just like most countries around the world, will

[1] https://laion.ai/blog/laion-5b/
[2] https://enterprisersproject.com/article/2020/10/artificial-intelligence-ai-ethics-14-statistics

http://dx.doi.org/10.21622/ACE.2026.06.1.1906

soon grapple with these issues, given that organizations in Kenya are quickly adopting AI to improve their efficiency and productivity, and to stifle competition. For instance, Worldcoin [9] was barred from collecting iris-biometric data, sensitive personal data, from Kenyan in exchange for financial incentives[3]. Another example is the "Huduma Namba" project [10], an initiative rolled out by the Government of Kenya to centralize a national digital identity database that was later halted by the courts in 2021[4]. In both cases, the court faulted data protection and privacy concerns [10].

Collectively, these illustrations demonstrate that the extent to which we can exploit AI technology in solving our societal problems and challenges is a direct function of how committed we are as a humanity in mainstreaming and integrating ethics, responsibility, and human-centric computing in AI. To achieve this, we also need to develop architectures that designate users (humans) with the responsibility for AI. Users should be able to exercise control, authority, and accountability across the entire AI lifecycle, including its design, development, implementation, deployment, and adoption. This will enable holding AI system owners and engineers accountable when their AI systems are not aligned with our ethics, morals, and socialization as a society [2]. This demonstrate why Kenya, similar to other African countries, may need to urgently develop a generation of scientists, engineers and technologists who are human centric AI enthusiasts – AI experts who are not only technologically adept but who are also ethically, morally and socially grounded and able to think and reason ethically, and champion for ethical AI technologies in the society [9].

It follows, therefore, that AI engineers and researchers need to understand ethical concerns of the AI systems that they develop right at the conception of the project, and then leverage ethical and responsible AI frameworks and guidelines to inform their decision on whether to implement or drop the project. This is an essential skill towards achieving ethical AI that needs to be embedded in CS.[5] curriculum whose graduates end up as AI practitioners and experts in the industry and academia. Evidently, this could ensure that the need for the creation of ethical and responsible AI systems is internalised in the mindset of the future AI practitioners and experts at the earliest possible time, thus assisting in creating a community of ethical AI activists and champions. The premise is that CS students are the future drivers of AI innovations and hence they need to be empowered to formulate, design, develop and deploy AI applications that are ethical. As such, they need to be equipped and motivated to promote ethical AI initiatives in their AI projects and to champion it as "a must have" and not just an afterthought or a "good to have". In other words, if society expects organizations to create ethical AI, there is no doubt that this desire will remain unattainable until organizations that create these AI systems are able to identify and onboard skilled ethical AI engineers and scientists. Conversely, where would these organizations recruit ethical AI engineers and scientists from unless universities train them? This demands that the universities redesign their AI course syllabi to integrate and train students on ethical and responsible AI as required by the job market. Evidently, such a training should develop capacity among the students to implement ethical AI in their projects, and to advocate for ethical AI systems at their future workplaces. This argument is similar to the assertion by [9, 11] that CS curriculum can only prepare its graduates with technical skills, as well as ethical and socio-cultural awareness, only if these non-technical skills are intentionally integrated into the curriculum. This ensures that graduates develop the competence to understand the potential legal, social, and ethical implications and consequences of their designs, alongside technical competence. According to [12], the CS curriculum needs to empower students to develop a strong understanding of the complexities of power and underrepresentation in computing and how AI can exacerbate these imbalances in the context of power and governance, ethics, equity, race, and even social responsibility.

According to [2], an ethical AI system can only be formulated, designed, or developed, and advocated by a team that is intentional on delivering an ethical AI system. And to achieve this intentionality, the team must have this concept intertwined in the fabric of their training and professional development, which can only be achieved from the very onset of their career development in AI. This means that every AI course in the CS curriculum should integrate

[3] High Court of Kenya. (2025). Republic v. Tools for Humanity Corporation (US) & 8 others; Katiba Institute & 4 others (Judicial Review Application No. E119 of 2023), [2025] KEHC 5629 (KLR). Kenya Law. https://new.kenyalaw.org
[4] High Court of Kenya. (2021). Nubian Rights Forum & 2 others v. Attorney General & 6 others [2021] KEHC 282 (KLR). Kenya Law. https://new.kenyalaw.org
[5] The Computer Science (CS) is used as an umbrella term referring to all disciplines of computing sciences such information technology, computer engineering, computer science, software engineering, information systems among others.

http://dx.doi.org/10.21622/ACE.2026.06.1.1906

ethical and responsible computing principles. In a typical CS curriculum [11], AI courses are taught separately from the general ethics ones, which mostly cover everything about professional ethics and legal issues in CS, with no specifics on the connection between ethics and CS principles, skills, and knowledge that the students cover in the CS curriculum. These courses on ethics are argued to appear as afterthoughts in the CS curriculum, and hence, students hardly give them much attention [13, 14]. A typical AI syllabus is, therefore, designed to scope and cover the core technical issues that the students need to learn to meet the learning expectations of the syllabus and assumes that ethics in AI is a peripheral non-issue which students will grasp from the professional ethics unit taught separately. As expected, this creates a disconnection between AI as a course and ethics in CS, as learners miss out on how to integrate ethical reasoning and thinking in AI projects [11, 15, 16]. Then it is impressive that to inculcate ethical AI in CS graduates, we must rethink and redesign the AI syllabus to integrate ethics, and teach it using a student-centered pedagogical framework that provides students with practical and real-world demonstrations.

1.1. CONTRIBUTION

In this paper, we describe an initiative proposed by the Department of CS at Kisii University to bridge the aforementioned gap by integrating ethical reasoning and thinking into the AI course syllabus in the CS curriculum and delivering it through a co-teaching approach with the Faculty of Law, the Faculty of Psychology, and ethical AI industry experts. Specifically, the paper reports on students' perception and experience on the proposed collaborative teaching of ethical AI course among undergraduate students of CS and how learning from this experience can be transferred to other undergraduate CS courses. Our aim is to advance our understanding of how to promote student-centered learning, encourage collaboration between academia and industry, and across disciplines in the university for CS curriculum delivery, and advance dialogue on the integration of ethics in the AI syllabus. Therefore, this paper interrogates two issues: (i) how can ethics be integrated in an AI syllabus and collaboratively taught by experts from the faculties of CS, law, and psychology, and ethical AI industry professionals; and (ii) what is the students' experience and perception in integrating ethics in an AI syllabus? The learning outcome indicates that the students appreciated that ethics in AI cannot be an afterthought after completing an AI project, but must be integral in AI system formulation, design, and implementation. Students reported that the learning sessions were engaging and interesting, and satisfactorily provided individual attention and learning support.

2. RELATED WORK

The emergence of AI technology is significantly changing how we live, work, interact, communicate, seek education, or health services, just to mention a few. As such, AI is becoming a core technology in our daily lives, a function that makes AI a technology that we cannot wish away. As the old aphorism goes, *"fire is a good servant but a bad master"*, so is AI technology, which has been shown to cause unintended but harmful consequences when its design, development, and deployment do not align with our ethics, morals, values, and culture as a society. Taxonomy developed by [17] classifies these unintended AI harm, risks and vulnerability as: adversarial threats; privacy and personal data protection violation; mis/disinformation, AI biases and discrimination; system safety and reliability failures; socioeconomic exploitation and inequality; carbon emission, environmental and ecological misuse; loss autonomy and weaponization; loss accountability, and human interaction and psychological harm.

Unfortunately, most of these harms [17, 18] only become evident after the AI system has been deployed to the market and not during the development. At this point, the developer's response becomes reactive, aimed at addressing these negative effects as activists and users demand accountability. In this case, ethical consideration is an afterthought being treated post-development of the AI system. Ideally, ethical thinking and reasoning should be integrated into the lifecycle of AI system development to enable early detection and treatment of potential harms. Achieving this, AI system owners and creators must develop

http://dx.doi.org/10.21622/ACE.2026.06.1.1906

a balance between maximizing return on investment and mitigating AI harm, and between pursuing efficiency, and ensuring inclusivity and representativeness in the datasets used, among others [19]. Nonetheless, at the center of this dilemma, what is fundamental is the ability to identify, reason through, and solve ethical problems in AI-based systems - a critical skill that CS departments must deliberately cultivate in CS graduates. According to [1], integrating ethical consideration in the development of an AI system must be sensitive to our cultural values, beliefs, and ethical principles. That is, at the core of an ethical AI system is social and cultural sensitivity and awareness to the diverse user population that it intends to serve.

For clarity, we define ethics in AI, similar to [14, 20], as concept and framework of social responsibility and social justice principled on social values, beliefs and moral guidelines that informs how AI systems can be formulated, designed and developed for societal good. Therefore, the social responsibility and framing of justice in the context of AI relates to aligning the development of AI systems to the ethical principles - fairness and non-discrimination, accountability, transparency, explainability, privacy and data protection, non-maleficence, inclusion and accessibility [21]. As such, ethics in computing education is intended to inculcate ethical and informed decision-making, and to leverage this reasoning in the creation of AI systems [15]. So, talking of ethical computing or ethical AI, the attention is on skills and practices that embed ethical thinking and reasoning into the formulation, creation, testing, deployment, and use of AI technology [16]. Given this discussion, we will use the ethical AI and responsible AI synonymously.

In integrating ethics in AI syllabus, the goal is to ensure that the AI syllabus meets and delivers the bare minimum of ethical (responsible) AI as per the above definition framework - with expectation that this contributes to building a community of graduates of CS who will palliate AI system harm and biases in the society by advocating for it through systems that they develop [20, 22, 23]. In addressing this need, universities have developed standalone computing ethics courses in the CS curriculum, while some have integrated ethics modules and discussions into pre-existing curricula [13]. For a while now, this has been a discussion in the community of practice of computer education on the role curriculum could play towards the attainment of an ethical AI system in society, and how this can be achieved. Such initiatives are exhaustively discussed and evaluated in [20] and [16].

According to [24], CS curriculum in most universities in Africa hardly integrates ethical computing, a reason we can attribute to the low ethical AI activism and championing from the AI professionals in Africa. This implies that while the job market would ordinarily expect AI graduates to have these ethical reasoning and thinking, and ethical dilemma-solving skills by the time they enter the job market, universities have not been responsive enough to redesign their syllabi to meet this need. Potentially, this has also created a skillset mismatch between the market demand and that possessed by CS graduates, which arguably contributes to an increase in youth unemployability in Africa. In this regard, authors such as [24, 25] call for urgent interrogation of how Faculty teach, how students learn, and what baselines of ethics are covered in AI and related courses, such as machine learning, data sciences, among others, in view of developing necessary polices and frameworks to support integration of ethical computing in syllabi of these courses.

Given the foregoing, the CS syllabus must be aligned accordingly to equip its graduates with an understanding that just like any other technology, AI is neither neutral nor does it exist in a vacuum. It contains within it inherent biases that reflect the perspective, preference, norm, culture, and perception of its creators and the data from which it is created. Specifically, such biases may arise from unrepresentative or incomplete training data, or due to the way the data was collected and sampled, so called data bias, or bias introduced by the design or assumptions from the AI algorithm itself, so called algorithmic bias, or bias reflecting the assumptions or perspectives of developers or dominant cultural norms, so called developer or cultural bias among other forms of biases [26]. In this regard, [26] draws our attention to the reality that technologies are hardly flawless and that it would be a mirage to purport that we can mitigate all ethical concerns in an AI system. Nonetheless, we have an obligation to optimize the societal benefits of AI while reducing its potential harm. This is in addition

http://dx.doi.org/10.21622/ACE.2026.06.1.1906

to developing these AI systems in an open manner, to the extent of sharing the model and data used to create them.

While it is evident from [13, 15] that most CS departments in the global north understand the value of integrating ethics in the CS curriculum, a couple of factors have been listed to derail this initiative. These issues include: Faculty not being in control of course curriculum and therefore does not have flexibility to adjust a course syllabus as and when they need; Faculty feeling that ethics is for other departments and does not have a place in CS curriculum; feeling that CS course content does not relate to ethics; feeling that technical concepts and skills in computing are of more importance than the ethical skills among others. To surmount these barriers, [13] proposes a collaborative approach built on inter-departmental and inter-disciplinary collaboration between the Faculty of CS and other faculties to co-teach some courses in the CS curriculum. Evidently, this can be enriched further by involving guest lectures from the ethical AI experts and professionals from the industry. They could deliver workshops and seminars on ethics and AI and develop learning resources, including case studies and reflections on practical ethical AI dilemmas that CS faculty may grapple with. In Section 3, we build on this discussion to propose a pedagogical framework to support CS faculty to integrate ethics in the AI syllabus and co-teach it with their colleagues from the faculties of psychology and law, and ethical AI industry professionals and experts. The framework is a multi-pedagogical strategy that incorporates the traditional content-based approaches, such as lecture methods, with emerging active learning approaches, such as problem-based learning, flipped classroom, and project-based learning. This builds on the Embedded EthiCS approach in [19] that limited the collaborative teaching to the faculties of CS and philosophy only. Different from the Embedded EthiCS approach, this paper proposes joint teaching with the ethical AI industry practitioners, which has the advantage of providing the students with real-world case studies of ethical AI dilemmas and reflections on how they handle them, and lessons learnt.

3. MATERIALS AND METHODS

This study adopted a design-based research methodology to enable us to iteratively co-design, co-develop, and co-evaluate interdisciplinary teaching materials and pedagogical frameworks collaboratively with the Faculty from departments of law, CS, and psychology, and ethical AI industry experts. This formed a team of eighteen experts – nine from the Faculty of CS, and two each from the faculties of law and psychology, and five ethical AI experts from the industry. The AI syllabus was redesigned to incorporate principles of ethics, ethical reasoning and thinking, and strategies for solving ethical dilemma problems. This resulted in an ethical AI (EAI) syllabus that was used to teach 130 third-year undergraduate students in the department of CS in the first semester of the 2024/2025 academic year at Kisii University. The team also co-developed a pedagogical framework for implementing this syllabus in addition to course materials such as course notes and tutorials, class and lab assignments, workshop and discussion tasks, guidelines for implementing the EAI class mini-project, and an end of semester exams.

During student onboarding, informed consent was obtained, and a pre-test was administered to assess students' understanding of ethical AI. Students then carried out the ethical AI mini-project in groups of five, under the supervision of a CS faculty member and an expert from either the faculties of law or psychology, or from industry.

Both qualitative and quantitative data were collected through students' feedback using surveys, AI mini-project implementation progress monitor, final presentation evaluation reports, and performance in exams and assignments. We also carried out focus group discussions with students, Faculty, and EAI industry practitioners, and collected reflection write-ups from the participants. Additionally, we also analysed teaching materials and the pedagogical framework developed. Qualitative data were analysed using content analysis to identify patterns related to learning experience and pedagogical effectiveness, while quantitative data were analysed using descriptive statistics to examine trends in student perceptions and learning experience.

http://dx.doi.org/10.21622/ACE.2026.06.1.1906

3.1. FROM AI SYLLABUS TO ETHICAL AI SYLLABUS

The AI course is a level 300 course taught to CS students aimed at introducing fundamental concepts, techniques, and applications of AI, covering the legacy and state-of-the-art AI technologies. At the end of the course, the students are expected to demonstrate theoretical understanding and practical ability to implement AI to solve a typical societal problem. Both the course objectives and content were redesigned to integrate the foundation and principles of ethics, and ethical reasoning and thinking to enable students to identify instances of ethical dilemma and to solve them, besides mastering the technical AI competencies. The structures for the AI course (before integration with ethics) and the revised AI course (after integrating with ethics, so-called ethical AI) are presented in Table 1.

Table 1: Course structures for AI and Ethical AI

Week	AI course outline	Ethics component integrated	Ethical AI course outline
1	Introduction to AI: definition, scope, history, and evolution of AI; strong AI vs weak AI; applications and limitations of AI	Benefits and harms of AI	Introduction to AI: definition, scope, history, and evolution of AI; strong AI vs weak AI; applications and limitations of AI; benefits and harms of AI
2	Intelligent agents: rational agents and environments; performance measures and rationality	Balancing utility functions vs human values	Intelligent agents: rational agents and environments; performance measures and rationality; balancing utility functions of AI vs human values
3 - 4	Problem solving and search: problem formulation; informed and uninformed search strategies; complexity analysis; heuristics and optimization	Optimization for whom -balancing efficiency vs fairness trade-offs; explore hidden assumptions in heuristics	Search and optimization: uninformed and informed search; heuristics and optimization; optimization for whom - efficiency vs fairness trade-offs; hidden assumptions in heuristics
5	Adversarial search and games: game playing and minimax algorithm; alpha-beta pruning; applications in strategic decision-making	Moral reasoning in adversarial contexts and responsibility in strategic system design – case of AI in conflict, surveillance, and warfare	Adversarial search: minimax algorithm; alpha-beta pruning; competitive AI vs cooperative outcomes; AI in conflict, surveillance, and warfare
6	Knowledge representation: logic-based representations; ontologies and semantic networks	Whose knowledge is represented? Cultural bias in ontologies and exclusion through formalization	Knowledge representation: logic-based representations; ontologies and semantic networks; cultural bias in ontologies and exclusion through formalization; inclusive knowledge modelling
7	Reasoning and inference: logical and rule-based inference; forward and backward chaining	Determinism vs human judgment; automation bias; when should humans override AI decisions?	Reasoning and inference: logical and rule-based inference; forward and backward chaining; determinism vs human judgment; human-in-the-loop reasoning
8	Uncertainty and probabilistic models: Bayesian reasoning; reasoning under uncertainty	Risk distribution across populations; thresholds and moral responsibility	Uncertainty and probabilistic models: Bayesian reasoning; reasoning under uncertainty; risk and harm anticipation
9	Machine learning: learning paradigms (supervised, unsupervised, reinforcement learning); model evaluation	Data bias and representation; consent and data provenance; historical bias in training data	Machine learning: learning paradigms (supervised, unsupervised, reinforcement learning); model evaluation; bias identification; ethical data practices
10	Machine learning algorithms: classification, regression, and clustering	Fairness metrics and trade-offs; comparative ethical reasoning	Machine learning algorithms: classification; regression; clustering; fairness metrics and trade-offs; comparative ethical reasoning; algorithmic biases
11	Neural networks and deep learning: neural architectures and model complexity	Transparency reasoning; opacity, explainability, and trust in black-box systems	Neural networks and deep learning: neural architectures and model complexity; opacity and explainability; transparency reasoning
12	AI mini-project: students formulate a basic problem and design and develop an AI solution for it.	End-to-end ethical evaluation of an AI solution.	AI mini-project: students formulate a basic problem and design and develop an AI solution for it; end-to-end integration of ethics in the AI solution

http://dx.doi.org/10.21622/ACE.2026.06.1.1906

In Table 1, the first column is the ordering of course content in the syllabus that covers 12 weeks of learning. The second column presents the course outline for the original AI course as offered by the department. The third column captures the ethics component that the faculties of CS, psychology, and law, and the ethical AI industry experts collectively identified for integration into the respective AI topics per week. The last column presents the Ethical AI course, that is, the AI course integrating ethics developed by the team from the brainstorming sessions – as shown in Figures 1 and 2. The continuous assessments – assignments, quizzes, group work, lab work, AI application mini-project, and exams all contributed to the final score for the course.

Figure 1: Team brainstorming session redesigning the AI curriculum for the level 300 CS curriculum to Ethical AI by integrating ethics, including ethical reasoning and thinking

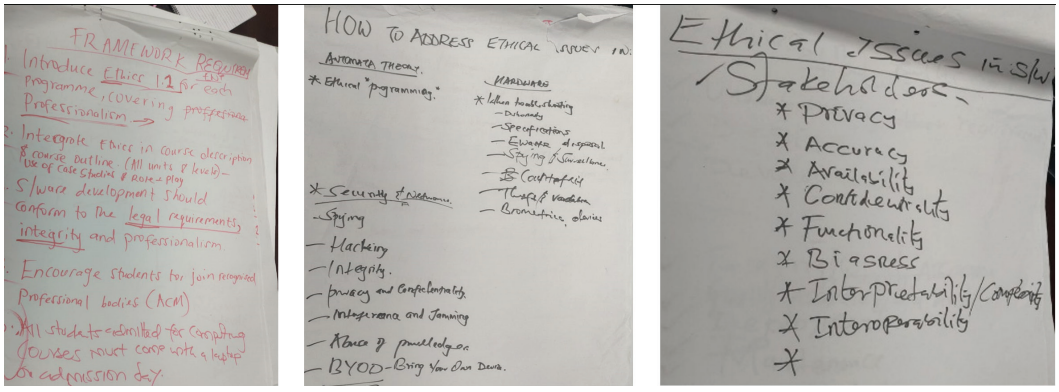


Figure 2: Sample of brainstorming output from the team

The proposed Ethical AI course aimed at integrating core concepts of ethics such as accountability, fairness, non-discrimination and bias, privacy and data protection, transparency and explainability, human in-the-loop/human oversight, AI harm prevention, inclusivity, and social impact into the AI syllabus. The premise here was that upon undertaking the redesigned Ethical AI syllabus, the students would be able to develop both technical AI competencies and skills, ethical responsibility, and professional judgement to align human values in AI systems that they create now and in the future.

3.2. FRAMEWORK FOR COLLABORATIVE TEACHING OF ETHICAL AI SYLLABUS

The team held multiple brainstorming sessions (see Figure 3) to formulate and design a co-teaching approach to structure and inform how the Faculty of CS will collaborate with their colleagues from the faculties of law and psychology, and the ethical AI industry experts and professionals in teaching the course.

http://dx.doi.org/10.21622/ACE.2026.06.1.1906

Figure 3: Experts from the faculties of CS, law, and psychology, and the ethical AI industry professionals, are brainstorming on the EAI collaborative teaching approach

The team proposed that the ethical AI course should adopt a student-centered approach that fosters active learning, enabling students to construct knowledge through discussions, problem-solving, mini-projects, and lab practicals, while guidance and support are provided by Faculty and industry experts. The approach was also designed to promote collaborative learning, including group work, peer-to-peer interactions, and co-teaching by Faculty across disciplines and industry professionals. To operationalize this, the team identified four essential components of the framework: problem-based learning, flipped classroom, case study and scenario-based learning, and project-based learning. These strategies were applied across lecture sessions, workshops, seminars, lab work, and project presentations, as summarized in Table 2.

Table 2: Learning approaches adopted

Learning approach	How to use	When to use	Who to use
Problem-based learning	To equip students with skills to integrate technical competence with ethical thinking and ethical reasoning. For instance, students are asked to analyse problems and identify technical and ethical issues in them, and to develop solutions.	During lectures, lab work, workshops, and seminar sessions	Faculty of CS, Law, Psychology, and Ethical AI industry experts
Flipped classroom	Students were provided with recorded lectures, reading assignments, and tutorials prior to class – these were then discussed in class.	During lectures and lab work	Faculty of CS
Case study and scenario-based learning	Students provided with practical, real-world AI ethical dilemma cases and scenarios - situations and experiences – to analyse and apply their knowledge to and come up with an ethically aligned and informed decision.	During the workshop and seminar, lectures, and lab work sessions	EAI industry experts, Faculty of Law and Psychology
Project-based learning	In groups of five, students develop an AI mini-project to solve a practical problem, considering ethical AI reasoning and thinking, and assessments.	During project consultation and presentation	Faculty of CS, law, psychology, and industry experts

By leveraging these four learning strategies, the teaching of ethical AI transitions from a traditional content-focused, faculty-led model to student-centered active learning. This approach encourages students to discover and construct knowledge through inquiry and reasoning, and to apply it in addressing real-world problems. It fosters a holistic learning environment that equips learners with both technical and ethical knowledge of AI, while providing opportunities to develop AI systems that are socially, culturally, and morally responsible. The proposed pedagogical framework, referred to as the CF2P framework, is presented in Figure 4.

To implement the ethical AI syllabus using the CF2P framework, the course was structured over twelve modules, comprising three lecture hours per week, two hours of lab sessions, one hour of weekly project consultation and reporting, two six-hour workshops, one three-

http://dx.doi.org/10.21622/ACE.2026.06.1.1906

hour seminar, and a dedicated day for the mini-project final presentation, totalling 103 contact hours. Lecture and lab sessions were facilitated by computer science faculty, while workshops and seminars were led by ethical AI industry experts and Faculty from law and psychology. Mini-projects were jointly supervised and evaluated by Faculty from CS, law, and psychology, alongside industry experts.

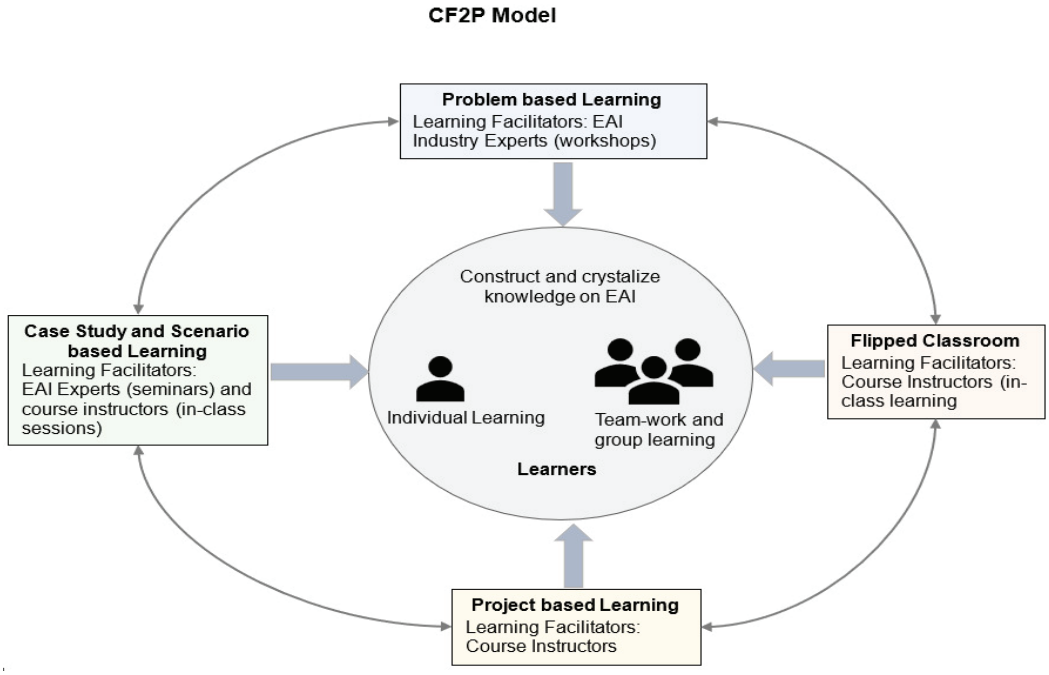


Figure 4: CF2P Pedagogical Framework

The proposed CF2P pedagogical framework engages CS faculty, Faculty from the departments of psychology and law, and ethical AI industry experts to co-deliver the syllabus, equipping learners with both technical competencies in AI and a deep understanding of ethics and moral responsibilities as AI practitioners. Collaboration with multidisciplinary Faculty and industry experts provided CS instructors with the necessary support to integrate ethics effectively into the course. This partnership also allowed Faculty to learn from peers in academia and industry, incorporating real-world case studies and documented experiences on ethical AI to enrich their broader teaching.

4. RESULTS AND DISCUSSION

The aim of this study was to redesign the Level 300 undergraduate AI course syllabus to incorporate principles of ethics, ethical reasoning, and critical thinking, and to implement it through a co-teaching approach involving industry experts and multidisciplinary Faculty from CS, law, and psychology. A summary of this implementation is presented in Table 3.

Table 3: Team co-teaching ethical AI course

SN	Description	Department	Number
1.	Internal Faculty	Computing Sciences	7
2.	External Faculty	Faculty of Law Faculty of Psychology	2
3.	External Faculty (Ethical AI industry experts)	Data Protection and Privacy Expert, Cybersecurity and AI Security Expert, AI Governance and Policy Expert, Human-Centered AI Expert, AI Product Manager, AI Ethics Lead	8
4	Students enrolled	Faculty of Computing Science	135

http://dx.doi.org/10.21622/ACE.2026.06.1.1906

4.1. LEARNING ETHICAL AI

At the start of the course, 55.6% of students reported being unfamiliar with ethical AI, and 84.3% indicated they had no prior experience working on or developing ethically-centered AI applications. Despite this, 96% of students agreed that AI systems should be ethical, yet 84.3% reported not knowing how to implement ethical considerations in their projects. Notably, 4% of students did not perceive a need for ethical AI systems. These findings suggest that the existing computer science curriculum primarily emphasizes technical computing skills, with limited focus on equipping students with the knowledge and capacity to be responsible and accountable for the societal impacts of the systems they develop. For some students, the responsibility for ethical computing appears externalized, considered an afterthought rather than an integral part of system design, consistent with observations reported in [11, 22, 27].

At the end of the course, 98% of students reported being familiar with ethics in AI systems, an increase from 44.4% at the start. However, only 55% indicated they could effectively integrate and apply ethical principles in a practical AI project, despite the course incorporating project-based and problem-based learning with mentorship in labs and workshops. Analysis of submitted projects revealed that approximately 45% did not adequately demonstrate understanding of key ethical AI principles, including accountability, explainability, transparency, data privacy and security, human agency, safety, fairness, and bias avoidance, all of which were covered in the course. This outcome can be attributed to the course assessment structure: class assignments (20%), continuous assessments (10%), a mini-class project (10%), and end-of-semester exams (70%). Consequently, students gave limited attention to the project component. Furthermore, as projects were completed in groups of five, it was observed that the majority of groups relied on one or two members with stronger programming skills, rather than collaborating equally on the project.

4.2. COLLABORATIVE TEACHING OF ETHICAL AI

At least nineteen out of twenty students, that is, 95.5%, reported being satisfied with the amount of individual attention and support provided during the learning sessions, indicating that the teaching approach ensured that all students participated and got involved in the learning. The students reported that the learning sessions were engaging and interesting, and that there were different and new learning activities, including guest lectures from industry experts, and the Faculty from the departments of law and psychology, and workshops and seminars that made learning fun. They also reported learning new soft skills such as teamwork, pitching and communication of science, human-centered AI system design and development, and presentation.

Their response also showed that case studies and scenario-based learning, and problem-based learning (30% and 28% respectively) were the most preferred learning approaches in the CS2P framework. They considered that these two approaches provided more engagement among students and between them and the facilitators. Other learning approaches that were reported to provide active participation were lectures and demonstrations (20.1%), followed by project-based learning (13%). The low preference for project-based learning could be attributed to low skill level in computer system project work, which is usually taught in year three, and as such, these students were learning computer system project work in a different course and were required to implement the skills in the ethical AI course. The flipped classroom was the least preferred learning approach by students (8.9%). This could be attributed to the need for pre-class personal engagement with course materials demanded in the flipped classroom approach that students found demanding. Notably, these different approaches being leveraged in this framework complemented each other, as 97% of the students appreciated that the teaching method used significantly enhanced their understanding and mastery of the course content. When asked about the likelihood of recommending this teaching method to their lecturers in other courses, 90% of the students indicated that they are likely to recommend this learning pedagogical framework. While students did not show a preference for any of the core components of the pedagogy if implemented as a standalone, we notice that combining them builds a holistic approach to

http://dx.doi.org/10.21622/ACE.2026.06.1.1006

learning and is preferred by the students. This is because the CS2P pedagogical framework leverages the strength of each individual learning framework to deliver a student-centric learning experience. To account for the 10% who did not find this approach of teaching to be effective, we argue that being a new approach, there is a need for mindset change among students and capacity building to promote acceptance and shift the mindset from the traditional lecture-based approach that they are used to. A summary of the activities carried out in this study and the results is presented in Table 4 below.

Table 4: Summary of the activities and results

Description	Results	Learnings	Challenges/Obstacles
AI course syllabus	An ethical AI course was developed by redesigning the AI course to integrate ethics.	The ethical AI syllabus is too broad to be covered in one semester – it can be divided into two levels, covering two semesters, to enable mastery of the content and practical learning.	Faculty of psychology and law needed to have some understanding of core concepts of AI to enable them to develop valuable feedback and insights on the integration of ethics in the AI syllabus.
Pedagogical framework for collaborative teaching	The student-centered co-teaching approach, the CF2P model, that integrates case studies, flipped classroom, project-based learning, and problem-based learning, was co-developed to co-teach the ethical AI syllabus.	The approach implements active learning and focuses the learning activity on the student. Preliminary results are promising, but need further investigations to see if this model can supplement the traditional lecture-based method of teaching in other computing courses, too.	Need to build the capacity of the Faculty on active learning teaching approaches. Need for mindset change from students and Faculty – for instance, students found the approach a little too involving, as they are used to the lecture approach that needed minimal involvement from them.
Learning materials, including recorded lessons, case studies, problems, etc	Course learning materials were developed, including assignments, case studies, video and audio recordings, group discussion tasks, tutorials, and lecture notes that students accessed from the university's learning management system.	Need for retooling and re-skilling of Faculty on instructional design and development of interactive content using the emerging authoring tools.	Most students could not afford Internet connectivity while off campus. Students relied on the university's internet connection for their learning.
AI class mini-projects integrating ethical AI concepts learnt	26 projects developed by the students, 55% of which integrated ethics – ethical reasoning and thinking in solving the problem using AI.	Ethical reasoning and thinking need to be integrated in all CS courses and class mini-projects.	Need for capacity building to CS faculty and students on the trade-off between the protection of intellectual property rights of students' innovations (projects) versus project mentorship by the faculty.

4.3. POTENTIAL GAINS

In collaborative teaching with the ethical AI industry experts and the faculties of law and psychology, the CS students reported having forged an inter-departmental and multidisciplinary network from these faculties and the industry, in addition to appreciating how ethical computing cuts across multiple disciplines. Students learnt so many other skills besides the AI and ethics skills – including marketing, communicating science in their project, communication skills, teamwork, pitching among others.

The diversity in the teaching approach and diversity in profiles of the Faculty and industry

http://dx.doi.org/10.21622/ACE.2026.06.1.1906

experts who delivered the course enabled the students to appreciate why ethics, morals, values, and culture are key in AI systems and need to develop AI systems with multidisciplinary experts. For instance, 93% of the students reported that the teaching method provided them with opportunities for collaboration and interaction with other learners in a manner they had never experienced before, and that they now appreciate the need for teamwork in developing computing applications even with peers from non-computing disciplines.

Additionally, majority of the students (87.6%) indicated that the ethical AI course made a great impact in their future career paths by motivating them to explore unexpected career paths such as ethical AI, AI safety and trust among others.

4.4. LESSONS LEARNED FROM INTEGRATING ETHICS IN AI

The baseline results reveal a significant gap in students' prior exposure to ethical AI. More than half of the students (55.6%) initially reported unfamiliarity with ethical AI, while 84.3% had no experience developing ethically centred AI applications. Despite this lack of exposure, a large majority (96%) acknowledged the necessity of ethical AI systems, though 84.3% did not know how to operationalize ethical principles in practice. This gap highlights a structural limitation in conventional CS curricula, which tend to prioritize technical competencies while giving limited attention to the societal and ethical implications of computing technologies. The perception among a small minority (4%) that ethical responsibility lies elsewhere further reinforces the notion that ethics has traditionally been treated as peripheral rather than integral to AI system development.

Following the Ethical AI course, the results demonstrated a substantial improvement in students' conceptual awareness of ethical AI. Familiarity increased to 98%, suggesting that the curriculum successfully introduced and contextualized ethical principles in AI development. However, the translation of conceptual understanding into practice remained limited, as only 55% of students demonstrated the ability to integrate ethical principles into AI projects. The analysis of submitted projects supports this finding, with approximately 45% failing to adequately demonstrate key ethical AI principles such as fairness, accountability, transparency, data privacy, and bias mitigation. This outcome appears to be influenced by assessment design, where the project component constituted only 10% of the overall grade and was completed in groups, often resulting in uneven participation.

The perception of students on the collaborative and interdisciplinary teaching approach was highly positive. The involvement of experts from the Faculty of CS, law, psychology, and industry enhanced engagement and broadened students' understanding of ethical AI as a multidisciplinary domain. High satisfaction rates (95.5%) and strong appreciation for interactive pedagogies - particularly case study/scenario-based and problem-based learning - indicate that such approaches effectively support deeper engagement with ethical considerations in AI.

The findings indicate that integrating ethics into AI education through collaborative and multidisciplinary teaching can enhance students' awareness and appreciation of ethical AI. However, the results also show that translating this understanding into practice remains a challenge. Strengthening assessment structures, particularly those that emphasize practical application, may help ensure that students are able to effectively integrate ethical principles into the design and development of AI systems.

While the results appear to provide insights on the integration of ethics in AI and collaborative teaching with interdisciplinary faculty and industry professionals, we appreciate that the study is limited to the context of Kisii University, with findings based on self-reported experiences and perceptions from students who registered for the course, which may be subject to response bias therein, affecting the generalizability of the findings. Subsequently, the responses may be influenced by response bias, which may limit the extent to which the findings may be generalized. Future studies could address this limitation by examining similar interventions across multiple institutions, courses, and contexts. Such broader investigations would allow for comparative analysis and provide stronger evidence regarding the applicability of the results in different educational settings.

http://dx.doi.org/10.21622/ACE.2026.06.1.1906

5. CONCLUSION AND FUTURE WORK

While society expects organizations to develop and deploy ethical AI systems, this goal is unlikely to be achieved without a pipeline of ethically trained AI graduates from universities. Consequently, universities must redesign their AI curricula to integrate ethics to producing graduates capable not only of formulating, designing, and deploying ethical AI systems but also advocating for responsible AI practices. To address this objective, this paper examines two questions: (1) how ethics can be integrated into the AI syllabus and collaboratively taught by Faculty from computer science, law, psychology, and industry experts, and (2) what would be the experiences and perceptions of students enrolled in an ethics-integrated AI course.

To this end, a team of Faculty from CS, law, and psychology, together with ethical AI industry experts, co-developed an ethical AI syllabus by integrating ethics into the Level 300 undergraduate AI course and implemented it for *135 CS students* using a novel *student-centered, collaborative* teaching framework. The aim was to cultivate AI graduates who understand that ethics is not an afterthought, but a core and integral component of AI system formulation, design, and implementation.

Preliminary results indicate that the proposed CS2P pedagogical framework can enhance student engagement and mastery of ethical AI concepts. According to student self-reports, the framework positions learners at the center of the educational process, with Faculty serving as facilitators rather than primary drivers, as in traditional lecture-based approaches. Most students (95.5%) found this approach satisfying, and 90% stated that they would recommend it for other computer science courses. By the end of the course, familiarity with ethics in AI increased from 44% to 98%.

These findings demonstrate that ethical principles, including reasoning and critical thinking, can be embedded within the core AI syllabus. Additionally, these show that learning can be enriched through collaborative teaching with the faculty of law and psychology, and ethical AI industry professionals providing students with a multi-faceted understanding of ethical AI. Moreover, students gained access to real-world case studies, networking opportunities, and potential collaborations with industry experts and Faculty across disciplines.

Looking ahead, it is essential to build the capacity of CS faculty to integrate ethical considerations into the core curriculum and to collaborate effectively with experts from other disciplines and industry in co-teaching. Achieving this requires a mindset shift among Faculty and motivation to embrace student-centered pedagogical approaches.

ACKNOWLEDGEMENT

This work was supported by the Mozilla Foundation through the Responsible Computing Challenge grant award to Kisii University. Our thanks to the faculty, industry experts, and students who assisted in the implementation of this intervention, with special thanks to Benard Maake, Ruth Chweya, Jane Maina, and Teresa Abuya for assisting in conducting the study.

All data and artifacts used in this study can be made available upon request.

REFERENCES

[1] E. Brynjolfsson *et al.*, "Canaries in the coal mine? six facts about the recent employment effects of artificial intelligence," *digitaleconomy.stanford.eduE Brynjolfsson, B Chandar, R ChenStanford Digital Economy Lab. Published August, 2025•digitaleconomy. stanford.edu*, 2025.

[2] D. O. Eke, S. S. Chintu, and K. Wakunuma, "Towards Shaping the Future of Responsible AI in Africa," 2023, pp. 169–193. doi: 10.1007/978-3-031-08215-3_8.

http://dx.doi.org/10.21622/ACE.2026.06.1.1906

[3] A. U. Karimy, J. Rasuli, and P. C. Reddy, "AI in Online Educational Access: Breaking Barriers for Female Learners," in *2024 21st International Conference on Information Technology Based Higher Education and Training (ITHET)*, IEEE, Nov. 2024, pp. 1–9. doi: 10.1109/ITHET61869.2024.10837681.

[4] R. Vyas, V. Kesharwani, C. Rohra, and R. Sundrani, "AI-Driven Platform for Direct Market Access for Farmers," 2025, pp. 713–730. doi: 10.2991/978-94-6463-738-0_57.

[5] R. Ladhar *et al.*, "AI-based Market Intelligence Systems for Farmer Collectives: A Case Study from India," *ACM Journal on Computing and Sustainable Societies*, vol. 1, no. 1, pp. 1–32, Sep. 2023, doi: 10.1145/3609262.

[6] M. C. Parlasca, C. Johnen, and M. Qaim, "Use of mobile financial services among farmers in Africa: Insights from Kenya," *Glob. Food Sec.*, vol. 32, p. 100590, Mar. 2022, doi: 10.1016/j.gfs.2021.100590.

[7] R. H. N. Sitinjak, S. M. A. Purba, S. Majidah, and N. A. Azis, "Towards a Non-Discriminatory Artificial Intelligence in Healthcare: Ensuring Equal Access and Utilization for Rural and Underdeveloped Areas," *Jurnal HAM*, vol. 16, no. 3, pp. 177–196, Dec. 2025, doi: 10.30641/ham.2025.16.177-196.

[8] J. Taylor, W. Agnew, M. Sap, S. E. Fox, and H. Zhu, "The Algorithmic Gaze: An Audit and Ethnography of the LAION-Aesthetics Predictor Model," *arXiv preprint arXiv:2601.09896*, 2026.

[9] C. Mbogho, S. Azeka, and J. Edward-Gill, "RESPONSIBLE COMPUTING IN KENYAN UNDERGRADUATE CURRICULA: INSIGHTS FROM A YEAR-LONG MULTIVERSITY INITIATIVE," Nov. 2025, pp. 7519–7528. doi: 10.21125/iceri.2025.2090.

[10] N. Kiilu, "Indirect Discrimination: Huduma Namba (Digital Identification) and the Plight of the Nubian Community in Kenya," *Strathmore Law Review*, vol. 7, no. 1, pp. 17–47, Oct. 2022, doi: 10.52907/slr.v7i1.188.

[11] L. Cohen, H. Precel, H. Triedman, and K. Fisler, "A New Model for Weaving Responsible Computing Into Courses Across the CS Curriculum," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, New York, NY, USA: ACM, Mar. 2021, pp. 858–864. doi: 10.1145/3408877.3432456.

[12] J. J. Ryoo and T. Blunt, "'Show Them the Playbook That These Companies Are Using': Youth Voices about Why Computer Science Education Must Center Discussions of Power, Ethics, and Culturally Responsive Computing," *ACM Transactions on Computing Education*, vol. 24, no. 3, pp. 1–21, Sep. 2024, doi: 10.1145/3660645.

[13] J. J. Smith, B. H. Payne, S. Klassen, D. T. Doyle, and C. Fiesler, "Incorporating ethics in computing courses: barriers, support, and perspectives from educators," in *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, New York, NY, USA: ACM, Mar. 2023, pp. 367–373. doi: 10.1145/3545945.3569855.

[14] G. Musumba, M. Kagiri, H. Mwangi, D. Wachepele, and D. Miano, "Embedded Responsible Computing: Mainstreaming Ethics into Computer Science Curriculum: A Case Study of Dedan Kimathi University of Technology (DeKUT)," in *2024 International Conference on Information and Communication Technology for Development for Africa (ICT4DA)*, IEEE, Nov. 2024, pp. 247–252. doi: 10.1109/ICT4DA62874.2024.10777260.

[15] S. Pasricha, "Ethics in Computing Education: Challenges and Experience with Embedded Ethics," in *Proceedings of the Great Lakes Symposium on VLSI 2023*, New York, NY, USA: ACM, Jun. 2023, pp. 653–658. doi: 10.1145/3583781.3590240.

[16] S. A. Doore, M. Trim, J. Streator, R. L. Blumenthal, A. , Rudra, and R. B. Schnabel, "Multiple Approaches for Teaching Responsible Computing," *arXiv preprint arXiv:2502.10856*, 2025.

[17] N. Seghid, F. Iqbal, K. Al-Room, and Á. MacDermott, "Emerging threats in AI: a detailed review of misuses and risks across modern AI technologies," *Frontiers in Communications and Networks*, vol. 6, Feb. 2026, doi: 10.3389/frcmn.2025.1727425.

http://dx.doi.org/10.21622/ACE.2026.06.1.1906

[18] D. Acemoglu, "Harms of AI," Cambridge, MA, Sep. 2021. doi: 10.3386/w29247.

[19] B. J. Grosz *et al.*, "Embedded EthiCS: integrating ethics across CS education," *Commun. ACM*, vol. 62, no. 8, pp. 54–61, Jul. 2019, doi: 10.1145/3330794.

[20] N. Brown, B. Xie, E. Sarder, C. Fiesler, and E. S. Wiese, "Teaching Ethics in Computing: A Systematic Literature Review of ACM Computer Science Education Publications," *ACM Transactions on Computing Education*, vol. 24, no. 1, pp. 1–36, Mar. 2024, doi: 10.1145/3634685.

[21] I. D. Raji, M. K. Scheuerman, and R. Amironesei, "You Can't Sit With Us: exclusionary pedagogy in ai ethics education," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, New York, NY, USA: ACM, Mar. 2021, pp. 515–525. doi: 10.1145/3442188.3445914.

[22] N. Garrett, N. Beard, and C. Fiesler, "More Than 'If Time Allows,'" in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, New York, NY, USA: ACM, Feb. 2020, pp. 272–278. doi: 10.1145/3375627.3375868.

[23] J. Saltz *et al.*, "Integrating Ethics within Machine Learning Courses," *ACM Transactions on Computing Education*, vol. 19, no. 4, pp. 1–26, Dec. 2019, doi: 10.1145/3341164.

[24] I. T. Sanusi and F. G. Deriba, "What do we know about computing education in Africa? A systematic review of computing education research literature," *arXiv preprint arXiv:2406.11849*, 2024.

[25] CUE, "Kisii University: Initiative to Promote Ethical Use of AI among Students puts Varsity on the Map," Mar. 2024, *Varsity Digest 1*. [Online]. Available: https://fliphtml5.com/yvqhd/oaxr/

[26] B. Hofmann, "Biases in AI: acknowledging and addressing the inevitable ethical issues," *Front. Digit. Health*, vol. 7, Aug. 2025, doi: 10.3389/fdgth.2025.1614105.

[27] T. S. Goetze, "Integrating Ethics into Computer Science Education," in *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, New York, NY, USA: ACM, Mar. 2023, pp. 645–651. doi: 10.1145/3545945.3569792.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

DETERMINATION OF PARAMETERS FOR TEST CASE OPTIMIZATION IN ADAPTIVE CUCKOO SEARCH TECHNIQUE: A FUZZY DELPHI ANALYSIS

James Maina Mburu[1], John Gichuki Ndia[2] and Samson Wanjala Munialo[3]

[1,2] Murang'a University of Technology, Department of Information Technology, Kenya.

[3] Meru University of Science and Technology, Department of Information Technology, Kenya.

Emails: { jmburu48@gmail.com, jndia@mut.ac.ke, smunialo@must.ac.ke }

Received 09 April 2026 - Accepted 01 June 2026 - Published 07 June 2026

ABSTRACT

Cuckoo Search Algorithm (CSA) is widely used in test case optimization due to its simplicity and ease of implementation. However, the performance of CSA and its adaptive variants mainly relies on parameter configuration. Current studies mainly rely on heuristic tuning, empirical experimentation, and problem-specific adjustments, with limited use of systematic parameter selection approaches, which may potentially affect the efficiency, reproducibility, and consistency of the technique. To address this gap, this study proposes a structured Parameter Selection and Validation Framework for identifying and confirming critical parameters for the development of the Enhanced Adaptive Cuckoo Search Technique (EACST). The framework combines literature-based parameter identification, expert evaluation, and Fuzzy Delphi analysis to determine and validate critical parameters. The study conducted an expert opinion survey involving 56 software testing professionals, where parameters were initially identified through an extensive literature review and subsequently refined based on expert feedback. These responses were analyzed using the Fuzzy Delphi Method, which incorporates a consensus threshold($d \leq 0.2$) and defuzzification to assess agreement and rank parameters. The findings indicate that all the identified parameters satisfied the threshold condition and achieved a consensus level above 75%, with an overall agreement of 91%, signifying strong expert consensus. All parameters were accepted (α-cut ≥ 0.5), demonstrating their suitability for inclusion in the proposed technique. These results establish a reproducible framework for parameter selection that supports exploration–exploitation balance in test case optimization. However, the study is limited to expert-based evaluation and requires further validation through implementation. The proposed framework integrates literature-based parameter selection with Fuzzy Delphi analysis, bridging the gap between theoretical identification and practical technique development.

Keywords: *Enhanced Adaptive Cuckoo Search Technique (EACST), Expert Consensus, Exploration–Exploitation Balance, Fuzzy Delphi Method, Model-Based Testing, Parameter Selection, Software Testing, Test Case Optimization.*

1. INTRODUCTION

Software testing is a process in software engineering that ensures the production of quality, reliable, and correct software. It typically entails three main phases, namely: test case production, test case selection, and test execution. The most critical stage of software development, which consumes resources in terms of time, cost, and effort, is the test case generation [1]. Subsequently, the design of efficient techniques for generating optimal test cases has attracted significant research attention.

Currently, metaheuristic optimization techniques are employed to Unified Modeling Language

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

(UML), a standard modeling approach applied to represent system structure and behavior, which has been widely used for automated test case production. These techniques aim to improve testing efficiency by intelligently exploring for optimal test cases within large solution spaces. Techniques based on the Cuckoo Search Algorithm (CSA) and its variants [2],[3], Genetic Algorithms (GA) [4], Genetic–Crow Search hybrids [5], and Firefly–Bee Colony hybrids [6] have recorded promising performance in improving test coverage and optimization outcomes. However, they still experience premature convergence or slow search progression. These drawbacks culminate from an imbalance between global search and local refinement mechanisms within the optimization process [7].

CSA is among the well-established metaheuristic techniques due to its simplicity, low parameter dependency, and ease of implementation. The technique applies Lévy flight-based randomization and a probabilistic replacement approach to explore the search space. Despite these gains, the CSA displays notable shortcomings, namely, susceptibility to local optima, inadequate exploitation capability, and relatively slow convergence rates in challenging problem domains [8]. Various CSA variants, such as Adaptive Cuckoo Search (ACSA) [2], Adaptive Cuckoo Search based on a Dynamic Adjustment Mechanism (ACS-DAM)[9], Multi-Strategy Adaptive Cuckoo Search (MSACS) [10], Enhanced Cuckoo Search Algorithm (ECSA) [11], and Enhanced Cuckoo Search Optimization with Opposition-Based Learning (ECSO-OBL) [12], have been proposed and rely primarily on parameter adjustment mechanisms such as step size and abandonment probability.

The adjustment of parameters is often insufficient to effectively maintain population diversity, achieve stable convergence, or prevent premature convergence to local optima. Consequently, maintaining a consistent balance between exploration and exploitation remains a significant challenge. Furthermore, integrating multiple adaptive strategies may increase algorithmic complexity and introduce greater parameter dependency, hence the need to develop an Extended Adaptive Cuckoo Search Technique (EACST).To develop a new technique, it is necessary to determine the parameters required for its design. According to Shadkam [13], the efficiency of metaheuristic techniques largely depends on the appropriate Selection and adjustment of parameters. However, existing studies lack a systematic framework for parameter selection, often relying on arbitrary choices that may lead to the omission of critical parameters needed for the development of efficient techniques. They largely rely on heuristic tuning, empirical experimentation, and problem-specific adjustments for parameter configuration, with limited use of a structured, validated parameter selection framework. These limitations may result in inconsistent optimization performance, reduced reproducibility, and lower reliability of the proposed techniques. Therefore, there is a need for a structured parameter selection and validation mechanism to identify critical parameters that support an effective balance between exploration and exploitation, thereby improving optimization efficiency and ensuring consistent performance of the technique.

This study implements a structured framework for the identification and validation of critical parameters for the development of EACST. The approach integrates literature-based parameter identification, expert evaluation, and the Fuzzy Delphi Method (FDM) to establish a reproducible and validated basis for parameter selection. By incorporating expert knowledge and fuzzy analysis, the approach aims to determine the parameters that can enhance optimization in test case generation.

The remainder of this paper is as follows: Section 2 presents related literature, Section 3 describes the research methodology, Section 4 presents the findings, Section 5 discusses the results, Section 6 presents the threats to validity, and Section 7 concludes the study with recommendations for future research.

2. LITERATURE REVIEW

This section reviews existing studies on CSA variants with emphasis on parameter configurations, optimization approaches, and reported challenges. The review provides a basis for understanding current advancements and identifying critical parameters relevant to the proposed EACST Parameter Selection and Validation Framework.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

Salgotra et al. [14] suggest a Self-Adaptive Cuckoo Search (SACS) that integrates a Gaussian bare-bones sampling mechanism with adaptive control of critical parameters, such as population size, search space, switch probability, and maximum iteration. The SACS was evaluated using benchmark optimization functions, where it demonstrated good performance, positioned first with a mean f-rank of 3.10. However, the Gaussian bare-bones sampling approach only enhances the exploration of the SACS and does not sufficiently support exploitation capability. Furthermore, the technique remains sensitive to parameter configurations, and the parameter settings were not systematically validated. In addition, the evaluation is mainly limited to benchmark problems, raising concerns about generalizability and practical applicability.

Also, hybrid and application-specific improvements have been discovered. Jeyaboopathiraja et al. [15] introduced an enhanced Adaptive Cuckoo Search Algorithm (ACSA) integrated with Artificial Neural Networks (ANNs) for cybersecurity applications, specifically for intrusion detection in cloud and IoT domains. The technique employs ACSA to enhance network routing and feature selection, while ANNs are used for classification of regular and malicious activities. The algorithm includes adaptive mechanisms such as an adaptive step-size adjustment, adaptive population control, and inertia weight to enhance search efficiency and convergence. The model was evaluated with a real-world cybersecurity dataset and demonstrated better performance in terms of accuracy, precision, recall, and F1-score compared to existing methods. Nonetheless, the combination of multiple adaptive strategies and learning models increases its complexity and establishes additional parameters that need careful configuration. Furthermore, the parameter configurations rely on predefined settings without a systematic parameter selection strategy. Moreover, the technique is mainly domain-specific to cybersecurity applications, which may constrain its generalizability to other optimization problems.

Likewise, Samal et al. [16] proposed an Adaptive Cuckoo Search (ACS) approach for optimal relay node placement in Wireless Body Area Networks (WBANs). In this approach, the step size is dynamically adjusted based on the fitness function to enhance convergence and avoid local optima. The technique optimizes multiple objectives, such as cost, coverage, energy consumption, delay, and load balancing, attaining optimal relay placement with less energy consumption by the 100th iteration. However, the technique is characterized by a large number of iterations to achieve optimal solutions. In addition, the use of a multi-objective fitness function increases computational complexity and sensitivity to parameter tuning. Moreover, parameter settings are mainly based on problem-specific configurations, which may reduce adaptability across different optimization scenarios. Furthermore, the evaluation is restricted to simulated WBAN environments with predefined conditions, which may limit generalizability to real-world applications.

Similarly, Yang et al. [9] proposed an Adaptive Cuckoo Search based on a Dynamic Adjustment Mechanism (ACS-DAM). In this approach, both discovery probability and step size are adaptively adjusted using exponential and logarithmic functions to maintain global exploration and exploitation capabilities. The technique was evaluated on 23 benchmark functions and demonstrated superior performance in terms of convergence speed and solution accuracy compared to standard CSA and several existing variants. It was further used to enhance Support Vector Machine (SVM) parameters, resulting in improved classification accuracy and faster convergence. Nevertheless, the adjustment approach relies on predefined mathematical functions and fixed parameter adjustment rules, which may restrict adaptability across diverse problem domains. Furthermore, the technique was primarily evaluated using benchmark functions and controlled datasets, raising concerns regarding its applicability and reproducibility in real-world optimization problems.

Likewise, Sahoo et al.[2] Suggest an Adaptive Cuckoo Search Algorithm (ACSA) for model-based test case production using a UML sequence model. The approach transforms a sequence diagram into a graph and employs ACSA to enhance test paths for improved coverage. Results indicate faster convergence, reaching optimal solutions at 100 iterations compared to 220 for standard CSA. Nevertheless, the technique requires a relatively large number of iterations to achieve optimal solutions, which may limit efficiency in

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

large applications. Furthermore, the technique relies on predefined parameters without a systematic parameter selection strategy. Although adaptive step size improves the balance between exploration and exploitation, its rule-based nature, rather than feedback-driven adaptation, may limit its effectiveness in large and complex search spaces. Moreover, the evaluation was conducted on a single case study, which may restrict its applicability to complex software testing scenarios.

Moreover, Gao et al. [10] introduced a Multi-Strategy Adaptive Cuckoo Search (MSACS) technique that combines several search strategies with dynamic selection mechanisms to enhance the balance between exploration and exploitation. The technique integrates crucial parameters such as population size, search space, fitness function discovery probability, step size, and strategy selection probability, which impact convergence behavior. The technique was evaluated on 24 benchmark functions and demonstrated enhanced accuracy and stability compared to existing techniques. However, the technique still requires a relatively large number of iterations (1000–2000) to achieve optimal solutions, which may limit efficiency. In addition, the incorporation of multiple strategies increases algorithmic complexity and parameter dependency, while parameter settings are mostly determined through trial-and-error tuning rather than a systematic selection approach. Furthermore, the evaluation is constrained to benchmark functions, restraining applicability to real-world optimization problems.

Likewise, Xiong et al. [3] suggested a Cuckoo Search algorithm based on a cloud model (CCS), where the step size factor is dynamically modified using fitness-based membership functions to enhance convergence and search accuracy. The technique integrates vital parameters such as discovery probability, step size factor, and cloud model parameters, which influence search behavior. The technique was validated on 25 benchmark functions and chaotic time-series prediction problems, demonstrating improved convergence speed, accuracy, and stability compared to several CSA and non-CSS algorithms. However, convergence remains dependent on parameter tuning, and the parameter configurations were not systematically validated. Moreover, the technique was mainly validated in simulation environments, which may limit efficiency and applicability in real-world optimization problems.

In addition, Abdulwahab et al. [17] suggested a Multi-Objective Binary Cuckoo Search Algorithm (MOBCSA) for feature selection in bioinformatics, enhancing both classification accuracy and the number of nominated genes. The technique combines parameters such as population size, abandonment probability, and maximum iterations, along with mechanisms such as an external archive and crowding distance to preserve diversity. The technique was evaluated on biomedical datasets, attaining classification accuracy between 92.79% and 98.42% and an average of 15.67 to 27.88 genes. Nevertheless, the technique establishes extra parameters and algorithmic complexity without systematic parameter validation. Also, much emphasis is placed on exploration, with relatively limited exploitation mechanisms, leading to an imbalanced search process. In addition, the evaluation is restricted to biomedical datasets, limiting applicability to real-world problems such as model-based test case optimization.

Similarly, Anwariningsih et al. [18] introduced an enhanced Cuckoo Search algorithm with adaptive parameters and hybrid distribution (Dynamic Hybrid Cuckoo Search with Contrast Limited Adaptive Histogram Equalization (DH-CSA-CLAHE)) for improving Contrast Limited Adaptive Histogram Equalization (CLAHE) parameters in medical image enhancement. The technique combines parameters such as adaptive discovery rate and step size with a hybrid distribution and combination of normal and uniform strategies to enhance the balance between exploration and exploitation. The technique demonstrates better performance in terms of image quality metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Mean Squared Error (MSE). However, the integration of dynamic and hybrid approaches increases algorithmic complexity. In addition, the balance between exploration and exploitation is guided by predefined parameter schedules, which may limit adaptability across diverse problem domains. Furthermore, the technique relies on predefined parameter configurations without a structured parameter selection and validation framework. Moreover, evaluation is limited to medical image processing applications, restricting generalizability to other domains.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

Likewise, Villanueva et al.[11] developed an Enhanced Cuckoo Search Algorithm (ECSA) incorporating Sobol sequence initialization and adaptive parameter control using cosine annealing to enhance population distribution and convergence efficiency. The technique adaptively adjusts core parameters such as discovery rate and step size, achieving up to a 30% improvement in mean fitness and outperforming the standard CSA in 11 out of 13 benchmark functions, with statistically significant differences at the 5% significance level. Conversely, the technique establishes additional approaches that increase its complexity and parameter dependency. In addition, the performance in discrete optimization problems is not statistically significant, limiting its applicability to different problem domains.

In addition, Meena et al. [19] proposed a Cuckoo Search Optimization (CSO) technique improved with fuzzy logic for influence maximization in dynamic social networks. The technique integrated parameters such as discovery probability, maximum iterations, population size, and dynamic seed size, local search, while using closeness and degree centrality with influence strength in a fuzzy-based update mechanism to improve solutions. The technique was evaluated on several real-world datasets and indicated good performance in terms of activated nodes compared to benchmark techniques. However, the technique incurs computational cost due to fuzzy logic and centrality calculations, specifically in dense networks. Although the balance between exploration and exploitation is implicitly achieved through the interaction of Cuckoo Search operations and fuzzy-based enhancement, the absence of explicit adaptive control limits the technique's ability to respond effectively to dynamic search conditions. Furthermore, the technique relies on predefined parameter settings and lacks a structured parameter selection and validation framework. Moreover, the technique is mainly evaluated in social network contexts, restricting its applicability to other domains.

Lastly, Reddy et al. [12] suggested an Enhanced Cuckoo Search Optimization with Opposition-Based Learning (ECSO-OBL) for optimal sensor node placement in wireless sensor networks to improve coverage and connectivity. The technique combines parameters such as step size, population size, discovery probability, mutation probability, maximum iterations, and objective function while incorporating opposition-based learning, adaptive step-size control, and particle swarm optimization to improve exploration and exploitation. The technique achieved a maximum coverage rate of 98.45% in 143 iterations and reduced the execution time to 2.85 seconds, outperforming current techniques. However, the approach establishes several adaptive mechanisms, increasing parameter dependency and band complexity without structured validation; although convergence is enhanced, it still requires a large number of iterations. Overall, the reviewed studies highlight common trends in parameter usage, design strategies, performance outcomes, and associated limitations, which are summarized in Table 1.

Table 1: Summary of Reviewed Studies

Study	Parameters	Design / Strategy	Results Achieved	Identified Gaps
Salgotra et al. [14]	Population size, switch probability, Maximum iterations, Lévy flight (λ), search space, objective function	Self-Adaptive CS (Gaussian bare-bones)	Best performance (f-rank = 3.10)	Parameter sensitivity, weak exploitation, limited to benchmarks, and a lack of structured parameter validation
Jeyaboopathiraja et al. [15]	Population size, step size, inertia weight, Lévy flight (λ), objective function, Maximum iterations	Hybrid ACSA + ANN	Improved accuracy, precision, and recall	High complexity; parameter dependency; domain-specific, limited systematic parameter selection strategy
Samal et al. [16]	Population size, step size, iterations, multi-objective fitness function, Lévy flight (λ), Maximum iterations	Adaptive CS (WBAN optimization)	Optimal placement; minimal energy at 100 iterations	High iterations; complexity; limited to simulation, parameter settings based on problem-specific adjustment

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

Yang et al. [9]	Population size, step size, discovery probability, Lévy flight (λ), objective function	ACS with dynamic adjustment	Improved convergence, accuracy, stability	Predefined control; limited adaptability; benchmark-based, fixed parameter adjustment strategy
Sahoo et al. [2]	Population size, step size, iterations, lévy flight, fitness function, search space, Maximum iterations	ACSA for UML test case optimization	Improved convergence at 100 iterations	Iteration-dependent; single case study; parameter reliance, limited parameter optimization strategy
Gao et al. [10]	Population size, step size, Maximum iterations strategy probability, Lévy flight (λ), objective function	Multi-strategy adaptive CS	Improved accuracy and stability	High iterations complexity; trial-and-error tuning, lack of systematic parameter selection
Xiong et al. [3]	Population size, discovery probability, step size, cloud parameters, objective function	Cloud-based CS (CCS)	Improved convergence and stability	Parameter-dependent; simulation-based evaluation, limited parameter validation
Abdulwahab et al. [17]	Population size, abandonment probability (pa), Lévy flight, maximum iterations, fitness function, search space	Multi-objective binary CS	High accuracy (92.79%–98.42%)	Too many parameters; weak exploitation; limited domain, parameter settings based on problem-specific adjustment
Anwariningsih et al. [18]	Population size, abandonment probability (pa), iterations, fitness function	Dynamic hybrid CS	Improved PSNR, SSIM, MSE	Complexity; predefined balance; domain-specific, lack of structured parameter selection
Villanueva et al. [11]	Population size, step size, pa, Lévy flight (λ), objective function, Maximum iterations	ECSA with Sobol + cosine annealing	30% fitness improvement; statistically significant	Complexity; parameter dependency; sensitivity to parameter settings
Meena et al. [19]	Population size, pa, iterations, fitness function, Maximum iterations, local search (centrality), search space	CS + fuzzy logic	Competitive influence maximization results	High computational cost; implicit balance; domain-specific, lacks structured parameter selection
Reddy et al. [12]	Population size, probability of abandoning, step size, local search, mechanism, maximum iterations, mutation probability, Lévy flight (λ), objective function	ECSO-OBL (CS + PSO + OBL)	98.45% coverage; convergence at 143 iterations;	High parameter dependency; complexity; High iterations; complexity; limited to simulation, absence of a structured parameter selection mechanism

2.1. SYNTHESIS AND RESEARCH GAP

The reviewed studies indicate that, despite the variety of application domains and algorithmic improvements, a common set of parameters is consistently used in CSA and its adaptive variants. Parameters such as step size, population size, maximum iterations, abandonment probability, Lévy flight, search space, local search, and objective function play a fundamental role in governing exploration–exploitation balance, convergence behavior, and solution quality. However, the Selection of these parameters in existing studies largely relies on heuristic tuning, empirical experimentation, and problem-specific adjustments. Furthermore, although key parameters are commonly reported, limited evidence exists regarding a systematic procedure for selecting and validating parameter values, potentially affecting reproducibility, consistency, and applicability across studies.

Furthermore, Cuckoo Search variants still exhibit premature or slow convergence due to limitations in effectively balancing exploration and exploitation. Although adaptive, hybrid, and multi-strategy techniques have enhanced search performance, the balance remains inadequately controlled or not fully dynamic, resulting in inefficient search processes and a comparatively large number of iterations required to attain best solutions. In addition, the combination of numerous adaptive and hybrid mechanisms has meaningfully increased algorithmic complexity and parameter dependency, making these techniques hard to configure, computationally expensive, and less practical for real-world domains.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

In addition, most advanced Cuckoo Search variants are mostly evaluated in simulation-based environments, with limited application in model-based test case optimization. Existing implementations are frequently limited to single UML models and lack combined multi-UML frameworks, thereby restricting test coverage and reducing practical applicability in real-world software systems. Likewise, performance evaluation is often grounded on isolated metrics, focusing either on effectiveness or efficiency rather than a comprehensive assessment. The dependence on benchmark functions and small-scale case studies further limits generalizability and reduces confidence in real-world deployment.

To address these challenges, this study proposes an EACST Parameter Selection and Validation Framework that incorporates literature-based parameter identification guided by criteria such as frequency of use, role in exploration–exploitation balance, and empirical performance impact. The framework is further reinforced through expert evaluation and analysis using the Fuzzy Delphi Method, providing a systematic and reproducible basis for the design of EACST.

3. METHODOLOGY

This study used a structured expert opinion survey to validate the proposed EACST parameters using the Fuzzy Delphi Method (FDM). FDM is an enhancement of the traditional Delphi technique that combines fuzzy logic to manage uncertainty and vagueness in expert opinions [20]. The overall validation process was conducted in four phases to guarantee that the parameters selected are theoretically founded, practically relevant, and reinforced by expert consensus. The whole validation workflow is demonstrated in Fig.1 (EACST Parameter Selection and Validation Framework).

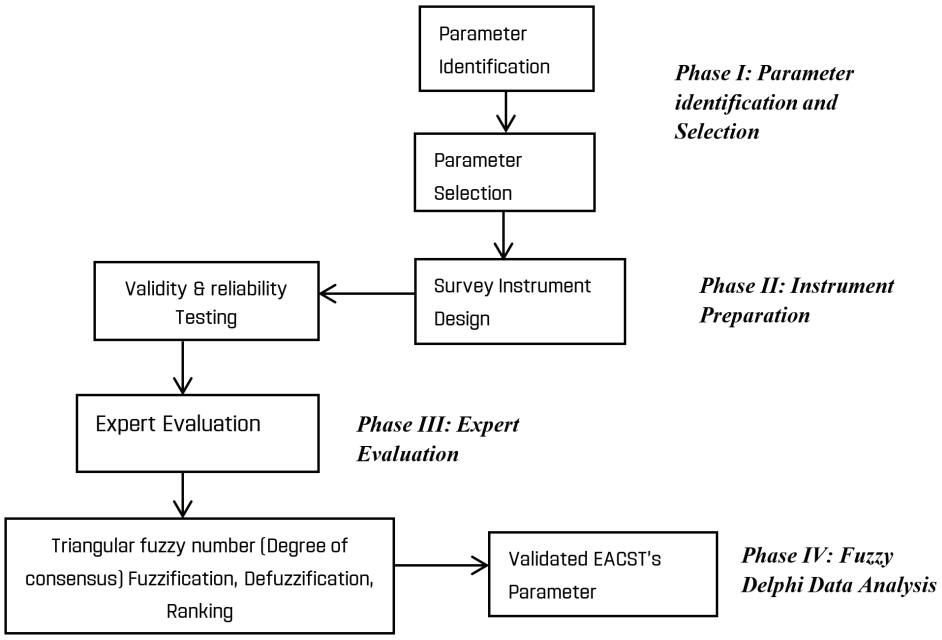


Figure 1: EACST Parameter Selection and Validation Framework

3.1. PHASE I: PARAMETER IDENTIFICATION AND SELECTION

This section describes the process used to identify and justify the critical parameters of the proposed Enhanced Adaptive Cuckoo Search Technique (EACST). The parameter identification process was guided by insights derived from reviewed studies on Cuckoo Search-based techniques to ensure that the selected parameters are supported by existing literature and practical applications. The process involves establishing selection criteria, mapping the identified parameters against these criteria, and providing justification for their inclusion in the proposed framework.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

3.1.1. Parameters Selection Criteria

The parameters of EACST were selected using a literature-driven approach based on three criteria, as shown in Table 2.

Table 2: Parameter Selection Criteria

Criterion	Description	Justification from Literature
C1: Frequency of Use	Parameter appears consistently across multiple ACS studies	Indicates maturity and wide research acceptance
C2: Role in Exploration–Exploitation Balance	The parameter directly influences global exploration or local exploitation behavior.	Essential for avoiding premature convergence
C3: Empirical Performance Impact	Parameter shown to improve convergence speed, solution quality, or coverage.	Supported by experimental evidence

3.1.2. Mapping of Parameters Selected

The mapping of selected parameters against these criteria (Table 3) shows that the majority satisfied all three conditions, indicating strong theoretical and empirical support in the literature. Parameters such as Population Size, Lévy Flight, Probability of abandonment, Search Space, Step Size, and Fitness/Objective Function satisfied all criteria. Maximum Iterations and Local Search Mechanism satisfied two of the three criteria but were reserved due to their demonstrated empirical impact on algorithmic performance.

Table 3: Mapping of Selected Parameters to Selection Criteria

Parameter	C1 (Frequency of Use)	C2 (Exploration–Exploitation Role)	C3(Empirical Performance Impact)	Ref
Population size (N)	✓	✓	✓	[2, 11, 12]
Maximum iterations	✓	–	✓	[10, 18, 19]
Abandonment probability (pa)	✓	✓	✓	[2, 10, 17]
Step size (α)	✓	✓	✓	[11, 16, 20]
Lévy flight (λ)	✓	✓	✓	[3, 16], 17]
Local search mechanism	–	✓	✓	[14, 17, 20]
Search space	✓	✓	✓	[17, 19, 20]
Fitness / objective function	✓	✓	✓	[2, 12, 16]

✓=Satisfy
–=Not Satisfy

3.1.3. Rationale for Parameter Selection

This section provides a parameter-specific justification for their inclusion in EACST. Each selected parameter is justified on grounds of its functional role in the optimization process and its empirical effectiveness as reported in prior studies. The rationale outlines how specific parameters contribute to search diversity, convergence behavior, and overall optimization efficiency, thereby supporting their integration into the proposed EACST framework. Table 4 maps the identified parameters to the relevant literature sources that support their inclusion.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

Table 4: Rationale for Parameter Selection

Parameter	Rationale from literature	Ref
Population Size (N)	Determines diversity of solutions; larger N improves exploration but increases computational cost.	[2, 3, 11, 12]
Max Iterations (I)	Defines a stopping criterion to prevent unnecessary computation.	[10, 18, 19]
Step Size (α)	Controls Lévy flight step length; adaptive adjustment improves convergence.	[14, 16, 20]
Abandonment Probability (pa)	Probability of replacing poor solutions; balances exploration and exploitation.	[2, 10, 17]
Lévy Flight (λ)	Enhances global exploration by allowing long jumps.	[2, 13, 16]
Local Search Mechanism	Refines promising regions for exploitation.	[14, 17, 20]
Search space	Defines the feasible region within which the algorithm operates	[17, [21], [12]
Objective Function	Guides the optimization process	[2, 12, 16]

The parameter determination procedure provides a systematic literature-driven basis for selecting the critical parameters of EACST. However, these parameters require validation to guarantee their relevance and practical applicability.

For that reason, the next section presents the methodology, where the identified parameters are operationalized into a survey instrument and confirmed using the Fuzzy Delphi Method to achieve expert consensus.

3.2. PHASE II: INSTRUMENT PREPARATION

To begin with, the identified parameters were converted into a semi-structured questionnaire for expert assessment. Every parameter was clearly defined and evaluated using a five-point Likert scale (Table 5) to preserve consistency in responses.

Table 5: Likert Scale

Scale	Response
1	Strongly Disagree
2	Disagree
3	Neutral
4	Agree
5	Strongly Agree

To ensure content validity, the instrument was checked by supervisors and three academic experts, who confirmed its appropriateness. In addition, a pilot study comprising nine software testing practitioners was performed to verify clarity and logical flow, leading to slight adjustments.

The reliability of the instrument was assessed using Cronbach's alpha, where a value of 0.70 or higher is generally considered acceptable [23],[24]. The instrument attained a coefficient of 0.939 across eight items, demonstrating a high level of internal consistency (as shown in Table 6.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

Table 6: Internal consistency

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
0.939	0.943	8

3.3. PHASE III: EXPERT EVALUATION

In this phase, software testers (experts) were chosen using a combination of purposive and snowball sampling. Purposive sampling ensured that only individuals with relevant expertise were included, while snowball sampling was applied to identify extra qualified experts through referrals. The validated questionnaire was subsequently distributed to the experts, who rated each parameter according to their level of agreement using a Likert scale. The responses received formed the base for determining the level of agreement among experts.

3.4. PHASE IV: FUZZY DELPHI ANALYSIS

The gathered responses were examined using the Fuzzy Delphi Method (FDM) to derive consensus on the suggested parameters. Each Likert-scale response was transformed into a Triangular Fuzzy Number where denote the least, most likely, and highest values, respectively. This conversion was conducted using the fuzzy scale presented in Table 7, which connects linguistic responses into equivalent fuzzy numerical values. This translation facilitates qualitative expert judgments to be conveyed quantitatively.

Table 7: Fuzzy Scale Agreement Levels

Response Category	Fuzzy Value Range
Strongly Disagree	(0.0,0.1,0.2)
Disagree	(0.1,0.2,0.4)
Neutral	(0.2,0.4,0.6)
Agree	(0.4,0.6,0.8)
Strongly Agree	(0.6,0.8,1.0)

To assess the level of consensus among experts, the threshold value was calculated using the distance between fuzzy numbers as shown in Equation (1):

$$d(m,n) = \sqrt{\frac{1}{3}\left[(m_1 - n_2)^2 + (m_2 - n_{2)}{}^2 + (m_3 - n_3)^2\right]} \quad (1)$$

A value of indicates satisfactory agreement among experts. Furthermore, a minimum consensus level of 75% was required for each parameter to be considered valid. These criteria ensure that only parameters with sufficient expert agreement are retained for further analysis [25],[26].

After evaluating expert agreement, defuzzification was performed to obtain an individual representative value for each parameter. This process transforms the Triangular Fuzzy Numbers into a crisp value to facilitate ranking and decision-making. In this study, the centroid method was applied as shown in Equation (2)[26]:

$$\mathrm{A} = \frac{1}{3} * (m_1 + m_2 + m_3) \quad (2)$$

The subsequent defuzzified value was then employed to determine the acceptance of each parameter. α-cut value of 0.5 was adopted as the decision threshold. Parameters with were acknowledged, indicating consensus among experts, while those with values below 0.5 were disallowed [20],[26]. Lastly, the parameters were ranked based on their defuzzified values to determine their relative importance within the proposed EACST framework.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

4. RESULTS

Throughout the application of the FDM, expert comments and proposals were integrated to enrich the parameters. After prioritization, the statements were reviewed, with no additional parameters proposed by the experts. Even though 56 responses were received, only 45 were considered valid, with 11 excluded due to incompleteness or duplication.

4.1. EXPERT PERSONAL INFORMATION

4.1.1. Academic Qualification

The academic qualifications of the experts who took part in the survey are shown in Fig.2. The majority of experts (71.4%) hold a Bachelor's degree, followed by 19.6% with a Diploma, while a smaller percentage (8.9%) hold a Master's degree.

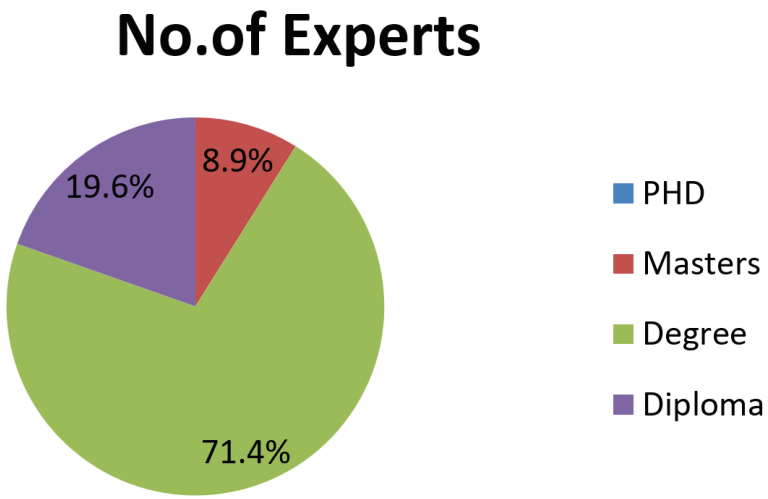


Figure 2: Level of education

4.1.2. Number of years in the software industry

Fig.3 shows the distribution of industry experience among the experts. The results demonstrate strong representation of experienced professionals. A substantial percentage (26.8%) has over eight years of experience, while 25% have 2–3 years of experience. Furthermore, 23.2% fall within the 4–5 year range, and 17.9% have 6–7 years of experience. Outstandingly, only 7.1% of the experts have less than two years of experience (0–1 year).

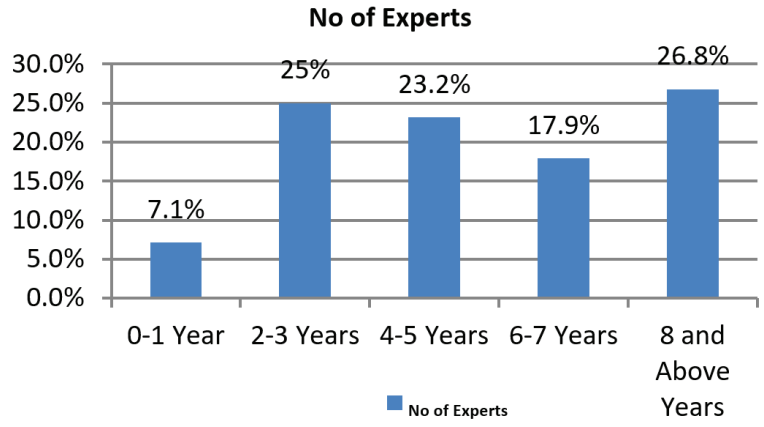


Figure 3: Number of years in the software industry

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

4.1.3. Knowledge in Software Testing

The results from Table 8 indicate that the largest number of experts ranked themselves between 3 (moderate knowledge) and 4 (high knowledge), with 4 being the most common rank.

Table 8: Knowledge in software testing

Rank	Frequency	Percentage (%)
1(Very Low)	0	0%
2 (Low)	5	11.1%
3 (Moderate)	15	33.3%
4 (High)	19	42.2%
5 (Very High)	6	13.3%
Total	45	100%

4.2. PARAMETER ANALYSIS

4.2.1. Expert Consensus Analysis

The findings of the Fuzzy Delphi analysis are illustrated in Table 9, which includes threshold value, percentage of expert consensus, fuzzy score, and parameter rankings. All parameters satisfied the threshold condition, demonstrating agreement among experts. Furthermore, the consensus level exceeded 75%, with an overall agreement of 91%, confirming strong expert consensus.

TABLE 9: Expert consensus about parameters of EACST

E	PS	I	LF	SS	PA	LS	OF	SP
1	0.2	0.4	0.2	0.4	0.2	0.2	0.2	0
2	0.2	0.2	0.2	0.2	0.2	0	0	0
3	0	0	0.2	0.2	0.2	0	0	0
4	0.2	0.2	0	0	0.2	0.2	0	0
5	0.2	0.2	0	0	0.2	0.2	0	0
6	0.2	0	0	0.2	0	0.2	0.2	0.2
7	0	0.2	0.2	0	0	0.2	0.2	0.2
8	0	0.2	0.2	0	0	0	0	0
9	0	0	0	0.2	0	0	0	0.2
10	0	0	0	0	0.2	0.2	0.2	0.2
11	0	0.2	0	0.2	0.4	0.2	0	0.4
12	0	0.4	0.2	0.2	0.2	0.2	0	0.2
13	0	0.2	0.2	0.2	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0.2	0.2	0.2	0.2	0	0.2	0.2	0.2
16	0.2	0	0.4	0.2	0.2	0.2	0	0
17	0	0.2	0	0	0	0.2	0.2	0.4
18	0.2	0.2	0	0.2	0.2	0.4	0.2	0.4
19	0.2	0.2	0.2	0.2	0	0.2	0.2	0.2

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

20	0.2	0.2	0.2	0.2	0	0.2	0.2	0.2
21	0	0.2	0	0.2	0.2	0	0	0.2
22	0	0	0	0.2	0.2	0	0	0.2
23	0	0.2	0.2	0	0	0	0.2	0
24	0.2	0.2	0	0	0	0	0	0
25	0.2	0	0	0.4	0	0.2	0	0.2
26	0.2	0.2	0	0.4	0	0.2	0	0.2
27	0.2	0.2	0	0	0	0.2	0.2	0
28	0	0.2	0	0.2	0	0.2	0	0
29	0.2	0	0.2	0	0.2	0	0	0
30	0	0	0.2	0	0	0	0	0
31	0	0	0	0	0.2	0.2	0.2	0.2
32	0	0.2	0	0	0.2	0.2	0	0
33	0	0.2	0	0	0	0	0.2	0
34	0	0.2	0	0	0.2	0.2	0.2	0
35	0.2	0.2	0.2	0	0.2	0.2	0.2	0.2
36	0.2	0.4	0.2	0.2	0.2	0.4	0.2	0.2
37	0.2	0.2	0	0.4	0	0.2	0.2	0
38	0.2	0.2	0.2	0	0	0.2	0	0.2
39	0.2	0.2	0.7	0.5	0.4	0	0	0.2
40	0	0	0.7	0.6	0.4	0	0	0.2
41	0	0.2	0	0	0.2	0	0	0
42	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
43	0.2	0	0.2	0.2	0	0	0	0
44	0	0	0	0.2	0	0.2	0.2	0
45	0.1	0	0.2	0.0	0	0.1	0.1	0.1
D	0.1	0.1	0.1	0.1	0.1	0.1	0	0.1
% CEEP	91%	93%	82%	89%	87%	93%	100%	96%
% CEAP				91%				
F	0.653	0.667	0.598	0.650	0.601	0.702	0.716	0.702
R	5	4	8	6	7	2	1	2

Key; E-Expert, **PS-**Population Size, **I-** No of Iterations, **LF-** Lévy Flight, **S-**Step Size, **P-**Probability of Abandoning, **LS-**Local Search, **OF-**Objective Function, **SP-** Search Space.
D-Threshold value, **CEEP-** Percentage Consensus of Experts for Every Parameter, **CEAP-** Percentage Consensus of Experts for all Parameters, **F-**Fuzzy Score, **R-**Parameter Ranking

The threshold values obtained for all parameters were within the acceptable limit, confirming consistency in expert opinions. In addition, the percentage consensus for individual parameters surpassed the required 75% threshold, while the overall consensus among experts reached 91%, showing strong agreement concerning the relevance of the proposed EACST parameters. These results indicate that the identified parameters satisfied the predefined criteria for subsequent ranking and acceptance analysis.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

4.2.2. Parameter Ranking and Acceptance

Following the evaluation of expert consensus, the validated parameters were ranked based on their defuzzified values to determine their relative importance within the proposed EACST framework. Parameters with values greater than 0.5 were accepted for inclusion in the technique. The ranking and acceptance results are presented in Table 10.

Table 10: EACST's Parameters Ranking Based on Fuzzy Score

Parameters	Fuzzy score (f)	Ranking (r)	Expert Consensus
Population Size(N)	0.653	5	Accept
Max. Iterations(I)	0.667	4	Accept
Lévy Flight	0.598	8	Accept
Step Size	0.650	6	Accept
Probability of Abandoning(pa)	0.601	7	Accept
Local Search Method	0.702	2	Accept
Objective Function	0.716	1	Accept
Search Space	0.702	2	Accept

The findings show that the objective function achieved the highest rankfollowed by search space and local search mechanisms . Lévy flight obtained the lowest score ; however, it still satisfied the predefined acceptance criteria. These findings confirm that all identified parameters are valid for inclusion in EACST.

4.2.3. Integration of Validated Parameters into EACST

The validated parameters were subsequently mapped into the proposed EACST framework to demonstrate their functional roles within the optimization process. As illustrated in Fig. 4, population size and search space will be used to generate and define the initial candidate solutions. The objective function will guide the optimization process by evaluating solution quality, while Lévy flight supports exploration, and step size regulates movement during the search process. Furthermore, the local search mechanism will facilitate exploitation of promising regions, abandonment probability will be applied to preserve population diversity, and maximum iterations will be used to define the stopping criterion for attaining optimal solutions.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

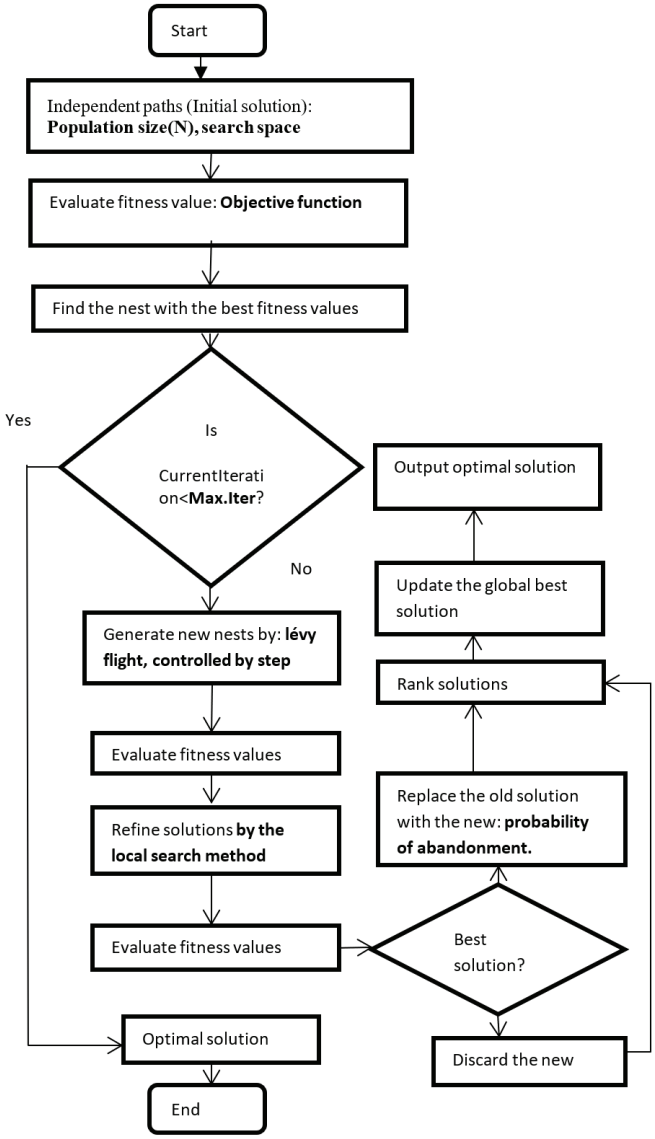


Figure 4: Parameter integration within EACST

5. DISCUSSION

5.1. EXPERT DEMOGRAPHIC DATA

The demographic findings demonstrate that the survey was conducted with a competent and experienced group of experts, guaranteeing the reliability of the results. The majority of respondents hold Bachelor's and Master's degrees, indicating a strong academic base in ICT-related fields, while their diverse industry experience offers a balanced blend of practical and theoretical perspectives. Furthermore, the reported moderate to very high knowledge in software testing approves the participants' capability to give informed evaluations. These characteristics reinforce the validity of the expert consensus and enhance confidence in the reliability of the parameter validation process, addressing common concerns in the literature regarding the subjectivity of expert-based methods.

5.2. PARAMETER ANALYSIS

The findings indicate that all proposed EACST parameters achieved strong expert consensus and satisfied the predefined acceptance criteria, confirming their relevance for adaptive

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

metaheuristic design. The high ranking of the objective function **(f = 0.716)** emphasizes its importance in guiding the optimization process through effective fitness evaluation. This observation is consistent with studies by Sahoo et al.[2] , Samal et al. [16] and Abdulwahab et al. [17], which identified the objective function as a major factor influencing convergence behavior and solution quality. The strong agreement among experts further demonstrates that an appropriate fitness evaluation mechanism is essential for determining the quality of candidate solutions and directing the search process toward optimal outcomes.

Similarly, the ranking of search space and local search mechanisms **(f = 0.702)** reflects the importance of exploitation strategies in refining promising solutions and improving convergence performance. Comparable observations were reported by Gao et al. [10] and Villanueva et al. [11], where adaptive local search mechanisms enhanced solution refinement and optimization efficiency. Defining an appropriate search space is equally important because it determines the region within which feasible solutions are explored and reduces unnecessary computational effort associated with irrelevant search areas. These findings suggest that effective exploitation strategies contribute significantly to improving optimization performance while minimizing premature convergence.

Parameters including maximum iterations, step size, and abandonment probability were also identified as important components influencing convergence behavior, search dynamics, and population diversity. Similar findings were reported by Yang et al. [9], who observed that adaptive adjustment of step size improved convergence speed and solution accuracy, and by Salgotra et al. [14], where adaptive parameter control enhanced search efficiency. Collectively, these parameters support the balance between exploration and exploitation, a concept widely recognized in metaheuristic optimization as a requirement for obtaining high-quality solutions while maintaining search efficiency.

Although population size and Lévy flight obtained comparatively lower rankings, their acceptance indicates that they remain important for preserving search diversity and strengthening global exploration capability. Similar observations were reported by Xiong et al. [3] and Anwariningsih et al. [18], where these parameters contributed to search-space exploration and prevention of premature convergence. Their relatively lower ranking may be attributed to expert perceptions that these parameters have a more indirect influence on optimization performance compared to parameters such as the objective function and local search mechanisms, which directly affect solution evaluation and convergence behavior.

The strong consensus achieved across all identified parameters suggests that a structured approach to parameter determination may improve consistency in the design of metaheuristic techniques. Existing studies often rely on heuristic tuning and problem-specific parameter adjustments, which can lead to variations in optimization behavior and reproducibility. The present findings show that expert-based validation provides a more systematic basis for identifying significant parameters, thereby supporting more reliable parameter configuration in adaptive Cuckoo Search techniques.

From a practical perspective, the validated framework may assist researchers and practitioners in configuring adaptive Cuckoo Search techniques with reduced dependence on trial-and-error parameter tuning. This can improve consistency in optimization performance and support more efficient test case generation within software testing environments. Nevertheless, the current findings are based on expert consensus and therefore require further experimental validation through implementation and performance evaluation of the proposed EACST in real-world software testing scenarios.

6. THREATS TO VALIDITY

6.1. INTERNAL THREATS

Sampling bias was a potential threat because access to industry experts was limited; therefore, snowball sampling was initially used to identify software testers through referrals from other qualified participants. However, snowball sampling may introduce bias because

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

participants are often recruited from existing professional networks, potentially excluding less-connected experts and limiting diversity within the sample. To minimize this threat, purposive sampling was employed to ensure that only participants with relevant expertise and experience in software testing were selected for inclusion in the study.

Instrumentation threat was considered because a poorly designed questionnaire could have misled experts and affected the accuracy of their responses. To minimize this threat, the questionnaire was reviewed by three academic experts who provided feedback and confirmed that the instrument met the required criteria. In addition, the instrument was tested for internal consistency using Cronbach's alpha, where the results indicated a high level of reliability.

6.2. EXTERNAL THREAT

Generalizability was considered a potential external threat because the findings of this study were derived from the opinions of software testing professionals. Therefore, the results may be applicable within software engineering contexts but may not fully represent perspectives from other domains, such as broader engineering fields or computer networks. Consequently, the applicability of the findings across diverse contexts should be interpreted with caution.

7. CONCLUSION AND FUTURE WORK

This study aimed to establish a structured and validated framework for identifying and validating critical parameters for the Enhanced Adaptive Cuckoo Search Technique (EACST) in test case optimization. The findings demonstrated strong agreement among experts, with all identified parameters satisfying the predefined acceptance criteria. Parameters, including objective function, search space, local search mechanism, maximum iterations, population size, step size, abandonment probability, and Lévy flight, were confirmed as important components for supporting effective and efficient test case optimization. The findings further indicate that optimization performance is influenced not only by the inclusion of these parameters but also by their systematic Selection and configuration.

The primary contribution of this study is the proposed EACST Parameter Selection and Validation Framework, which integrates literature-based parameter identification, expert evaluation, and Fuzzy Delphi analysis into a structured and validated process. Unlike many existing studies that rely on heuristic tuning and problem-specific parameter settings, the proposed framework provides a reproducible and systematic basis for parameter determination. This contributes toward improving consistency, reliability, and optimization efficiency in adaptive Cuckoo Search techniques while strengthening exploration–exploitation balance during optimization. From a practical perspective, the validated framework provides guidance for researchers and practitioners in configuring adaptive metaheuristic techniques for software testing applications. Nevertheless, the results are based on expert consensus and therefore require further validation through implementation and performance evaluation within real-world environments.

Future work will focus on the design, implementation, and evaluation of EACST software testing scenarios. Further refinement of parameters and integration with other optimization approaches may improve scalability, robustness, and applicability across broader optimization domains.

ACKNOWLEDGMENT

We extend our heartfelt gratitude to the panel of experts who participated in the survey.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

REFERENCES

[1] A.Tamizharasi , P. Ezhumalai, S. Remya Rose, P. Suresh, and S. Logesswarie, "Bio Inspired Approach for Generating Test data from User Stories," *Turkish Journal of Computer and Mathematics Education (TURCOMAT),* vol. 12, no. 2, pp. 412–419, Apr. 2021, doi: https://doi.org/10.17762/turcomat.v12i2.826.

[2] R. K. Sahoo, S. Satpathy, S. Sahoo, and A. Sarkar, "Model driven test case generation and optimization using adaptive cuckoo search algorithm," *Innovations in Systems and Software Engineering*, vol. 18, no. 2, pp. 321–331, Jan. 2021, doi: https://doi.org/10.1007/s11334-020-00378-z.

[3] Y. Xiong, Z. Zou, and J. Cheng, "Cuckoo search algorithm based on cloud model and its application," *Scientific Reports*, vol. 13, no. 1, pp. 1–13, Jun. 2023, doi: https://doi.org/10.1038/s41598-023-37326-3.

[4] R. K. Sahoo, M. Derbali, H. Jerbi, D. Van Thang, P. Pavan Kumar, and S. Sahoo, "Test Case Generation from UML-Diagrams Using Genetic Algorithm," *Computers, Materials & Continua,* vol. 67, no. 2, pp. 2321–2336, 2021, doi: https://doi.org/10.32604/cmc.2021.013014.

[5] A. Tamizharasi and P. Ezhumalai, "Genetic-based Crow Search Algorithm for Test Case Generation," *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies*, vol. 13, no. 4, pp. 1–11, 2022, doi: https://doi.org/10.14456/ITJEMAST.2022.74.

[6] S. S. Panigrahi, P. K. Sahoo, B. P. Sahu, A. Panigrahi, and A. K. Jena, "Model-driven Automatic Paths Generation and Test Case Optimization Using Hybrid FA-BC," *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pp. 263–268, Mar. 2021, doi: https://doi.org/10.1109/esci50559.2021.9396999.

[7] L. Liu, X. Liu, N. Wang, and P. Zou, "Modified Cuckoo Search Algorithm with Variational Parameters and Logistic Map," *Algorithms*, vol. 11, no. 3, p. 30, Mar. 2018, doi: https://doi.org/10.3390/a11030030.

[8] A. Sharma, A. Sharma, V. Chowdary, A. Srivastava, and P. Joshi, "Cuckoo Search Algorithm: A Review of Recent Variants and Engineering Applications," *Studies in Computational Intelligence,* pp. 177–194, Oct. 2020, doi: https://doi.org/10.1007/978-981-15-7571-6_8.

[9] Y. Yang, M. Fu, S. Yu, C. Jia, and X. Zuo, "Adaptive Cuckoo Search Algorithm Based on Dynamic Adjustment Mechanism," 電腦學刊, vol. 32, no. 5, pp. 171–183, Oct. 2021, doi: https://doi.org/10.53106/199115992021103205014.

[10] J. Cheng and Y. Xiong, "Multi-strategy adaptive cuckoo search algorithm for numerical optimization," *Artificial Intelligence Review,* vol. 56, no. 3, pp. 2031–2055, Jun. 2022, doi: https://doi.org/10.1007/s10462-022-10222-4.

[11] M. A. Villanueva, C. M. Ching, and K. Mata, "An Enhancement of Cuckoo Search Algorithm for Optimal Earthquake Evacuation Space Allocation in Intramuros, Manila City," *arXiv (Cornell University),* Feb. 2025, doi: https://doi.org/10.48550/arxiv.2502.13477.

[12] M. R. Reddy, M. L. R. Chandra, and R. Dilli, "Enhanced Cuckoo Search Optimization with Opposition-Based Learning for the Optimal Placement of Sensor Nodes and Enhanced Network Coverage in Wireless Sensor Networks," *Applied Sciences*, vol. 15, no. 15, p. 8575, Aug. 2025, doi: https://doi.org/10.3390/app15158575.

[13] E. Shadkam, "Parameter setting of meta-heuristic algorithms: a new hybrid method based on DEA and RSM," *Environmental Science and Pollution Research*, vol. 29, no. 15, pp. 22404–22426, Nov. 2021, doi: https://doi.org/10.1007/s11356-021-17364-y.

[14] R. Salgotra, U. Singh, S. Saha, and A. H. Gandomi, "Self adaptive cuckoo search: Analysis and experimentation," *Swarm and Evolutionary Computation*, vol. 60, p. 100751, Aug. 2020, doi: https://doi.org/10.1016/j.swevo.2020.100751.

http://dx.doi.org/10.21622/ACE.2026.06.1.2069

[15] J. Jeyaboopathiraja, P. Mariajohn, S. S. Maidin, and J. Sun, "An Adaptive Cuckoo Search Algorithm with Deep Learning for Addressing Cyber Security Problem," *Journal of Applied Data Sciences*, vol. 5, no. 4, pp. 1977–1988, Dec. 2024, doi: https://doi.org/10.47738/jads.v5i4.366.

[16] T. K. Samal, S. C. Patra, and M. R. Kabat, "An adaptive cuckoo search based algorithm for placement of relay nodes in wireless body area networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 1845–1856, May 2022, doi: https://doi.org/10.1016/j.jksuci.2019.11.002.

[17] H. M. Abdulwahab, S. Ajitha, A. Naji, B. Abdullah, and F. A. Ghanem, "MOBCSA: Multi-Objective Binary Cuckoo Search Algorithm for Features Selection in Bioinformatics," *IEEE access*, vol. 12, pp. 1–1, Jan. 2024, doi: https://doi.org/10.1109/access.2024.3362228.

[18] S. H. Anwariningsih, Wahyono, and R. Sumiharto, "An Improved Cuckoo Search Algorithm with Dynamic Parameters and Hybrid Distribution for Enhanced CLAHE," *Engineering, Technology & Applied Science Research*, vol. 15, no. 4, pp. 24148–24158, Aug. 2025, doi: https://doi.org/10.48084/etasr.10652.

[19] M. Sharma and B. Pathik, "Crow Search Algorithm with Improved Objective Function for Test Case Generation and Optimization," *Intelligent Automation & Soft Computing*, vol. 32, no. 2, pp. 1125–1140, 2022, doi: https://doi.org/10.32604/iasc.2022.022335.

[20] N. K. Ismail, S. Mohamed, and M. I. Hamzah, "The Application of the Fuzzy Delphi Technique to the Required Aspect of Parental Involvement in the Effort to Inculcate Positive Attitude among Preschool Children," *Creative Education*, vol. 10, no. 12, pp. 2907–2921, 2019, doi: https://doi.org/10.4236/ce.2019.1012216.

[21] S. K. Meena, S. S. Singh, and K. Singh, "Cuckoo Search Optimization-Based Influence Maximization in Dynamic Social Networks," *ACM Transactions on the Web*, vol. 18, no. 4, pp. 1–25, Oct. 2024, doi: https://doi.org/10.1145/3690644.

[22] S.-Y. Yang, Y.-H. Xiang, D.-W. Kang, and K.-Q. Zhou, "An Improved Cuckoo Search Algorithm for Maximizing the Coverage Range of Wireless Sensor Networks," *Mağallaẗ baġdād li-l-ʿulūm,* vol. 21, no. 2(SI), pp. 0568–0568, Feb. 2024, doi: https://doi.org/10.21123/bsj.2024.9707.

[23] H. C. Dandekar, "Qualitative methods in planning research and practice," *Journal of Architectural and Planning Research*, vol. 22, no. 2, pp. 129–137, 2005.

[24] S. Hari Sugiharto, "Comparative Test of Cronbach's Alpha Reliability Coefficient, Kr-20, Kr-21, And Split-Half Method," *Journal of Education Research and Evaluation*, vol. 8, no. 1, pp. 47–57, Feb. 2024, doi: https://doi.org/10.23887/jere.v8i1.68164.

[25] T. T. Yin and H. Hanif, "Fuzzy Delphi Method: A Step-by-Step Guide to Obtain Expert Consensus on MUETBot Functionalities," *International journal of academic research in business & social sciences*, vol. 14, no. 4, pp. 1097–1104, Apr. 2024, doi: https://doi.org/10.6007/ijarbss/v14-i4/21307.

[26] S. N. A. Mohamad, M. A. Embi, and N. Nordin, "Determining e-Portfolio Elements in Learning Process Using Fuzzy Delphi Analysis," *International Education Studies*, vol. 8, no. 9, Aug. 2015, doi: https://doi.org/10.5539/ies.v8n9p171.

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

MULTIMODAL MACHINE LEARNING MODEL FOR DETECTING KISWAHILI HATE SPEECH ON SOCIAL MEDIA

Banchale Adhi Gufu[1], Edward Ombui[2] and Audrey Mbogho[3]

[1] United States International University, School of Science and Technology, Kenya, Africa.
[2] United States International University, Artificial Intelligence , Kenya, Africa.
[3] United States International University, Machine Learning , Kenya, Africa.

Emails: { agufu@usiu.ac.ke, eombui@usiu.ac.ke, ambogho@usiu.ac.ke }

Received 10 April 2026 - Accepted 01 June 2026 - Published 08 June 2026

ABSTRACT

The rapid growth of social media platforms has transformed communication and information sharing across the world. However, it has also increased the spread of harmful online content, particularly hate speech. This problem is more critical in low-resource languages such as Kiswahili, where limited annotated datasets, language resources, and computational tools make it difficult to develop effective automated detection systems. Although many previous studies have focused on text-based hate speech detection in high-resource languages, hate speech on social media is often multimodal, combining both textual and visual elements such as images and memes. This study, therefore, solved this problem by developing a multimodal machine learning model for detecting hate speech in Kiswahili social media content using both text and image data. The study was guided by the research question: How can multimodal machine learning improve hate speech detection in Kiswahili using text and image data from social media platforms? To address this question, the research focused on several objectives, including the collection and annotation of a multimodal dataset, the development and evaluation of machine learning models, and the validation of the developed system using quantitative evaluation metrics and expert feedback. A dataset containing 115,204 Kiswahili text samples and 2,607 image samples was collected from social media platforms such as X and Facebook. Annotation was carried out on the preprocessed 112,571 texts and 2607 images, with the support of Kiswahili language experts to ensure linguistic and contextual accuracy. The study adopted a mixed research design, combining qualitative insights into hate speech interpretation with quantitative machine learning techniques. To address class imbalance in the training data, the Synthetic Minority Oversampling Technique (SMOTE) was applied only to the training set, while the validation and test sets were retained as representations of real-world data.

The dataset was divided using an 80:10:10 ratio, and hold-out validation was used to ensure reliable evaluation results. Several machine learning and deep learning models were developed and evaluated. The Bidirectional Long Short-Term Memory (BiLSTM) model achieved the best performance in text classification with an F1-score of 97.33%, while the ResNet50V2 model achieved 89.76% F1-score in image classification. The developed multimodal fusion model achieved an overall F1-score of 92.11%, demonstrating the effectiveness of combining textual and visual features in improving contextual understanding of hate speech. Although the text-only model achieved a slightly higher F1-score, the multimodal model provided a more robust and context-aware approach, especially in identifying implicit and visually encoded hate speech. The study contributes to the field by providing one of the first large-scale multimodal Kiswahili hate speech datasets and demonstrating the applicability of deep learning techniques in low-resource language settings. In addition, the study developed an interactive hate speech detection interface called ChujaHate, which was validated by experts. The findings support the development of more inclusive and culturally relevant content moderation systems while highlighting the importance of ethical AI considerations such as fairness, bias mitigation, and data privacy. Future research should explore

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

transformer-based multimodal architectures, real-time deployment, and the integration of additional modalities such as audio and video in dynamic online environments.

Keywords: *Hate speech detection, Late fusion, Low-resource languages, Machine learning, Multimodal data, Kiswahili, Natural Language Processing (NLP)*

1. INTRODUCTION

The rapid expansion of social media platforms has significantly transformed global communication, enabling individuals, communities, and institutions to interact and share information in real time across geographical boundaries. Platforms such as Facebook and X (formerly Twitter) have created open and participatory digital spaces that support collaboration and information exchange. However, alongside these benefits, social media has also facilitated the rapid spread of harmful content, particularly hate speech [1][2][3][4]

Hate speech has become a growing concern in online environments due to its potential to incite discrimination, hostility, and violence. The United Nations defines hate speech as any form of communication that attacks or uses discriminatory language toward a person or group based on identity factors such as Religion, ethnicity, nationality, race, or gender [5]. The viral nature of social media amplifies the reach and impact of such content, often intensifying social divisions and contributing to real-world conflict.

In many regions, including East Africa, hate speech has been linked to politically sensitive events such as elections, where it can fuel polarization and violence [6][7][8][9]. The increasing volume and speed of content generation make manual moderation ineffective, highlighting the urgent need for automated and scalable detection mechanisms.

To address the challenges associated with online social media hate speech, researchers have increasingly turned to automated detection systems based on machine learning techniques [10]. Machine learning enables systems to learn patterns from large datasets and make predictions with minimal human intervention [11][12]. Over time, hate speech detection has evolved from rule-based approaches to more sophisticated machine learning, deep learning models, transformer-based, and large language models.

Early studies relied on traditional machine learning algorithms such as Support Vector Machines (SVM) and logistic regression, which utilized manually engineered features like Bag-of-Words and Term Frequency Inverse Document Frequency (TF-IDF) representations [13][14]. While these approaches provided a foundation, they often struggled to capture contextual nuances and implicit forms of hate speech.

The introduction of deep learning techniques, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks, significantly improved the ability to model sequential and contextual information in text [15]. More recently, transformer-based architectures such as BERT and its multilingual variants, such as SwahBERT [16], have further enhanced performance by capturing deeper semantic relationships [17][18]. Despite these advancements, most research has focused on high-resource languages such as English, where large annotated datasets are readily available, leaving low-resource languages underrepresented.

Low-resource languages face significant challenges in the development of natural language processing systems due to the limited availability of annotated datasets, linguistic resources, and computational tools [19]. Despite Kiswahili being spoken by over 200 million people across Africa, it remains under-represented in machine learning research [20]. Existing studies on Kiswahili hate speech detection have primarily focused on text-based approaches using machine learning and deep learning models such as SVM, CNN, and LSTM [9][21]. While these studies demonstrate promising results, they are limited by small datasets and a narrow focus on textual data.

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

Additionally, Kiswahili presents unique linguistic challenges, including dialectal variation, code-switching with English, and evolving slang, which complicate the development of robust models [6][22]. These factors often lead to reduced model generalization and performance, particularly when applied to diverse real-world contexts.

The lack of large-scale, diverse, and well-annotated datasets further exacerbates these challenges, underscoring the need for more comprehensive resources tailored to the Kiswahili language.

A key limitation of existing hate speech detection systems is their reliance on unimodal data, particularly text. However, modern social media communication is inherently multimodal, combining text, images, videos, and audio to convey meaning [23].

Hate speech is often expressed implicitly through the interaction of multiple modalities. For instance, an image may contain symbolic or contextual cues that, when combined with accompanying text, convey a harmful message that cannot be detected through text analysis alone. This makes unimodal detection approaches insufficient for capturing the full spectrum of online hate speech.

Recent studies have demonstrated the effectiveness of multimodal approaches in improving detection accuracy [24][23]. For instance, [25] combined text and image features using BERT and ResNet models, achieving high performance. Similarly, [26] showed that fusion strategies integrating multiple modalities outperform unimodal baselines. Despite these advancements, multimodal hate speech detection remains largely unexplored in low-resource languages such as Kiswahili, where most existing research is still limited to text-based analysis [27][28]. Datasets play a critical role in the development and evaluation of machine learning models. However, the availability of high-quality datasets for hate speech detection in Kiswahili remains limited. Existing datasets are often small, domain-specific, and predominantly text-based [1][9]. The process of creating annotated datasets in low-resource languages is complex and resource-intensive. It requires linguistic expertise and cultural understanding to ensure accurate labeling, particularly for implicit and context-dependent forms of hate speech [29][30][31]. Recent efforts such as AfriSenti and AfriHate have contributed to the development of multilingual datasets for African languages, but these remain largely unimodal and do not capture the multimodal nature of social media content [32][33].

The absence of publicly available multimodal datasets for Kiswahili significantly limits the development of robust detection systems and hinders comparative evaluation across studies. This gap highlights the need for large-scale, culturally contextualized datasets that integrate both text and image data.

Given the limitations of existing approaches, there is a growing need for advanced solutions that integrate multimodal data and address the challenges associated with low-resource languages. Multimodal machine learning provides a promising framework for combining textual and visual information to improve detection accuracy and contextual understanding [34].

This study addressed these gaps by curating a multimodal Kiswahili hate speech dataset and developing a machine learning model that integrated both text and image data from social media platforms such as Facebook and X. The approach leveraged deep learning architectures, including Bidirectional LSTM for text processing and convolutional neural networks for image analysis, combined through a late fusion strategy to generate a unified prediction.

By capturing complementary information across modalities, the model can detect both explicit and implicit forms of hate speech more effectively than unimodal systems. In addition, the study incorporated ethical considerations such as bias mitigation, data privacy, and fairness in model development [35][36].

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

This study does not primarily introduce a new deep learning architecture but rather addresses an important research and resource gap in low-resource African language processing through multimodal integration. The main contribution of the study lies in the development of a large-scale multimodal Kiswahili hate speech dataset and the implementation of a multimodal framework that integrates textual and visual information for hate speech detection in social media contexts. Unlike many prior Kiswahili hate speech studies that rely solely on textual analysis, this work demonstrates the practical applicability of multimodal machine learning in capturing implicit and context-dependent hate expressions communicated through both language and imagery. The study further contributes to comparative benchmarking across classical machine learning, deep learning, and multimodal approaches within a low-resource setting.

2. RELATED WORK

In Africa, hate speech has emerged as a significant challenge, contributing to ethnic polarization, social unrest, and threats to democratic processes [6][7][8]. The rapid growth of social media platforms has further intensified this problem by enabling the widespread dissemination of inflammatory and harmful content in real time. Consequently, the development of automated hate speech detection systems has become increasingly important in supporting peaceful ethnic discourse and safeguarding social cohesion.

Early studies on hate speech detection primarily focused on text classification approaches using both traditional machine learning and deep learning techniques. Commonly applied methods include Support Vector Machines (SVMs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks. For instance, [37] demonstrated the effectiveness of feature-engineering approaches, particularly TF-IDF combined with psychosocial features, in detecting hate speech in Kiswahili text. Similarly, [1] reported that transformer-based architectures such as mBERT and XLM-RoBERTa outperform traditional models in code-switched English-Swahili datasets. In another notable contribution, [6] developed a Swahili and code-switched English-Swahili political hate speech dataset by integrating the Politikweli, AfriSenti, and Hate_Speech_Kenya datasets. Their comparative evaluation showed that non-linear SVM, Bi-LSTM with fastText embeddings, and SwahBERT achieved strong classification performance.

Despite the valuable contributions of these studies, most existing approaches remain limited to textual analysis, overlooking other modalities through which hate speech is communicated. Recent advancements in artificial intelligence and natural language processing indicate a growing transition from unimodal text-based systems to multimodal approaches that integrate text, images, audio, and video data [27][23]. This transition is motivated by the observation that hate speech on social media is often implicit, contextual, and distributed across multiple modalities, especially in memes, edited images, and multimedia posts where harmful intent may not be directly expressed in text alone.

Multimodal approaches seek to capture complementary information from different data sources [38], thereby improving the ability of models to identify hidden or implicit hate speech patterns [39]. For example,[25] proposed a multimodal framework that combined BERT for textual representation and ResNet for image feature extraction, achieving high classification performance with an accuracy of 97.0% and an F1-score of 94.7%. Similarly, [40] developed a transfer learning-based multimodal framework known as TCAM using the Facebook Hateful Memes dataset and demonstrated the importance of modeling weak semantic relationships between textual and visual modalities.

Further studies have continued to demonstrate the effectiveness of multimodal learning strategies in hate speech detection. For instance, [41] evaluated several multimodal datasets and models and reported that early fusion techniques combined with transformer-based architectures such as CLIP, UNITER, and BERT consistently outperform unimodal baselines on benchmark datasets, including MMHS150K and MAMI. Likewise, [38] introduced a transformer-based multimodal framework incorporating multi-head attention mechanisms,

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

achieving strong generalization performance across multiple hate speech datasets. In addition, [42] emphasized the importance of explainability and interpretability in multimodal hate speech detection systems, particularly for enhancing transparency and trustworthiness in automated moderation systems.

Although multimodal hate speech detection has gained considerable attention in high-resource languages, research in African and other low-resource languages remains limited and underexplored. Most studies involving Kiswahili and related African languages continue to focus primarily on text-based approaches. Examples include [37], which employed SVM-based classification using psychosocial features, and [1], which applied transformer-based models to code-switched English–Swahili text classification. More recently,[33] introduced the AFRIHATE dataset covering 15 African languages, including Kiswahili, where transformer-based models such as AfroXLMR achieved a strong performance of up to 91.72% in Swahili hate speech detection. However, the AFRIHATE dataset remains predominantly text-based, highlighting a critical limitation in current African hate speech research.

Only a few studies have attempted multimodal hate speech detection within low-resource African contexts [39]. For example, [27] developed a multimodal hate speech detection model for Amharic memes using CNN-based text and image fusion techniques, achieving approximately 81% classification accuracy. Similarly, [28] introduced a multimodal Urdu-English hate speech dataset and demonstrated that early fusion techniques significantly improve detection performance. Additionally, [22] and [43] explored audio-based hate speech detection in Kiswahili and English contexts, respectively, further demonstrating the potential of multimodal methods in low-resource environments.

Despite these advancements, research on multimodal hate speech detection in African languages remains fragmented, limited in scope, and underdeveloped. A review of recent studies, as summarized in Table 1, reveals that the majority of existing works are predominantly text-based, with very limited multimodal research focusing on African languages, particularly Kiswahili. This study, therefore, addresses this critical gap by developing a multimodal Kiswahili hate speech dataset sourced from X (formerly Twitter) and Facebook and a multimodal machine learning framework that integrates both textual and visual information. In doing so, the study contributes to the advancement of multimodal natural language processing research for low-resource African languages and provides a foundation for more robust hate speech detection systems in multilingual and multicultural online environments.

Table 1: Summary of key findings from the literature

Study	Dataset size and language focus	Best model	Metrics	Findings	Gaps
Transformer-based multi-task learning for multimodal hate speech detection (memes) By [26]	Four English benchmark meme datasets: MMHS150K, Facebook Hateful Memes, MultiOFF, SemEval MAMI	Multi-task learning framework combining UNITER + CLIP + BERT	AUC-ROC 0.817, Acc 77.35; MultiOFF: F1 79.82, Acc 81.54 (selected headline results)	Shows that training related meme tasks jointly can transfer useful multimodal knowledge and outperform several unimodal, fusion-based, and pretrained V+L baselines on multiple benchmarks.	Limited exploration of culturally diverse and low-resource contexts in multimodal settings; lacks real-world validation.
Swahili and code-switched English-Swahili political hate speech detection textual dataset by [6]	Merged Politikweli, AfriSenti, Hate_Speech_Kenya -101,014 tweets; includes hate/offensive/neither subset (19,369 labeled as hate/offensive/neither reported) Focused on Swahili + English–Swahili code-switching (Kenyan social media)	SwahBERT fine-tuning (plus BiLSTM and SVM baselines)	SwahBERT F1 = 0.89; BiLSTM (fastText) Acc/Prec/Rec/F1 = 0.80; recall for hate improved 0.57→0.76 with SMOTE	Contributes a larger, better-annotated Swahili/code-switched resource with language and target labels, and demonstrates that Swahili-specific pretraining (SwahBERT) can separate hateful from non-hateful content effectively despite imbalance.	Does not incorporate multimodal data, such as images, video, audio, among others, and is limited to political text.

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

AfriHate: multilingual hate/abusive datasets for 15 African languages (incl. Swahili) By [33]	Train/dev/test counts per language (e.g., Swahili: train 14,760; dev 3,164; test 3,168). Labels: hate, abusive, neutral; targets annotated Focused on 15 African languages (incl. Swahili)	Baseline comparisons: Africa-centric PLMs, SetFit few-shot, and prompted LLMs	Average macro-F1: AfroXLMR-76L ≈ 78.16; GPT-4o: ≈61.89 (zero-shot) and ≈70.79 (20-shot)	Provides a comparatively large, standardized multilingual benchmark with native-speaker annotation and explicit ethical guidance, enabling more realistic evaluation across African languages.	Focused only on text and lacks multimodal datasets and unified benchmarks.
Transfer learning + cross-attention/ cross-mask for hateful meme classification (FHMC) By [39]	FHMC: 12,140 unique meme images; official splits used (train 8,500; dev_seen 500; test_seen 1,500) with additional unseen splits reported. Focused on English	TCAM (with CLIP + TweetEval encoders; cross-attention + cross-mask fusion)	AUROC 0.8362; Accuracy 0.764 on FHMC test_seen	Shows gains from explicitly modeling weak/indirect relationships between meme text and imagery, outperforming several reported baselines and approaching top competition systems while remaining simpler than ensemble-heavy approaches.	Evaluated mainly on a single dataset; it lacks generalization and robustness testing.
Comparative study of transformer models for English–Kiswahili code-switched hate speech detection by [1]	HateSpeech_Kenya (Kaggle): 48,057 instances (Hate 3,181; Offensive 8,543; Neither 36,333). Focused on English–Kiswahili code-switched texts.	XLM-RoBERTa (best on balanced data); also mBERT/ mDistilBERT as feature extractors and end-to-end	Balanced: XLM-R Acc 0.6069; Macro-F1 0.49. Feature extractor best: mBERT-SVM Acc 0.5461; Macro-F1 0.40. Imbalanced: mBERT Acc 0.7820; Macro-F1 0.53	Highlights that end-to-end fine-tuning generally beats using transformers only as feature generators, but results are sensitive to class balancing choices, underscoring the need to report macro metrics and dataset conditions clearly.	Performance remains relatively low and shows unresolved issues in code-switching understanding.
ASR vs acoustic word embeddings for hate-speech keyword spotting in Wolof/Swahili radio By [22]	Scraped Swahili radio broadcasts from three Kenyan stations (out-of-domain); hate keywords labeled by native Swahili experts Focused on: Wolof + Swahili (speech)	Keyword spotting via fine-tuned multilingual ASR (XLS-R) vs ASR-free multilingual AWE (query-by-example)	In-the-wild precision (top-100 retrievals): ASR 30h=52%, 1h=42%, 5min=36%; AWE=45% (music retrievals far lower for AWE: 2% vs 17–30%)	Shows that even when ASR is sample-efficient in controlled settings, ASR-free embeddings can be more robust under real broadcast conditions, making them attractive when labeled in-domain data are scarce.	Keyword-based detection lacks deep semantic/contextual understanding of hate.
Amharic hate speech detection from Facebook memes using deep learning (OCR-based) By [27]	5,000 Facebook text-images (memes); split using StratifiedKFold with 80/10/10 (train/val/ test) Focused on Ahmaric	BiLSTM + Dense (two hidden layers)	Best accuracy ≈ 81%	Demonstrates a workable pipeline for a low-resource language by extracting meme text via OCR and training sequence models, indicating that reasonable performance is possible with modest data sizes.	Relies on OCR text only; ignores visual features in memes.
Multimodal hate speech detection in Greek social media (rendered tweets) By [25]	Collected ~126,000 tweets; labelled sample n=4,004 (1,040 toxic; 2,964 non-toxic); rendered tweet screenshots used for vision input Focused on Greek (with some English tweets retained)	Early fusion approach: fine-tuned BERT (text) + ResNet (image) jointly	Accuracy 0.970; F1 0.947 (best model)	Introduces a practical way to treat posts as they appear to users (rendered screenshots), capturing visual cues that plain text pipelines can miss, and reports strong performance on their Greek xenophobic/ racist dataset.	Potential dataset bias and lack of cross-domain validation.
Psychosocial features for hate speech detection in code-switched texts (Kenya elections) By [37]	≈50k human-annotated tweets (Kenya 2012 & 2017 elections); each tweet annotated by ≥3 annotators; 80/20 split with cross-validation Focused on English/ Swahili/Sheng + some local-language tokens	SVM with psychosocial (PDC) features (plus lexical/PoS/app features explored)	SVM overall accuracy ~76.2%; hate-class precision 0.81, recall 0.85, F1 0.83; also notes uniform P/R/F1 ≈0.77 for SVM in one evaluation summary	Shows that engineered psychosocial/topic-informed features can improve detection of camouflaged hate in code-switched settings, where purely lexical keyword approaches often miss nuanced attacks.	Focused only on English, Kiswahili conde switched text and lacks multimodal datasets and unified benchmarks for Kiswahili, feature engineering not.

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

3. METHODOLOGY

A mixed research design approach was used in which qualitative socio-cultural insights were integrated into the quantitative machine learning pipeline to enhance construct validity and annotation consistency. The architectural model presented in Figure 1 provides a structured framework for detecting Kiswahili hate speech by integrating both textual and visual information within a unified multimodal system. The process begins with the collection of 115,204 texts and 2,607 image data from social media platforms such as X (formerly Twitter) and Facebook, followed by annotation to label content as hate or non-hate speech. The collected data is then preprocessed separately according to modality. Text preprocessing involved cleaning, normalization, tokenization, stop-word removal, and conversion into numerical representations using Word2Vec embeddings. Similarly, image preprocessing included resizing, rescaling, augmentation, and feature extraction to prepare visual data for learning.

A structured annotation process of cleaned 112,571 texts and 2,607 images was conducted to ensure the reliability and validity of the multimodal dataset used in this study. Seven annotators, who were native Kiswahili speakers and active social media users, participated in the annotation exercise over a one-month period. Prior to annotation, the annotators underwent training and orientation sessions where they were introduced to the study objectives, labeling categories, and detailed annotation guidelines. The annotation process involved labeling both text and image data, with overlapping subsets assigned to multiple annotators to assess consistency and reliability. Regular discussions and review sessions were conducted to resolve disagreements and clarify ambiguous cases, ensuring a consistent interpretation of hate speech within the Kiswahili social media context. The study adopted a dual annotation scheme consisting of a nine-class labeling system for text data and a binary hate and non-hate classification system for image data. The text categories included not hate, offensive, sexual, gender, disability, race, chronic_disease, Religion, and Tribe, enabling fine-grained classification of different hate speech forms. Annotation guidelines emphasized context sensitivity, target identification, distinction between offensive and hateful content, handling ambiguity, and multimodal interpretation of text-image relationships. To evaluate annotation reliability, fifty text samples were independently labeled by multiple annotators and analyzed using Randolph's free-marginal multirater Kappa. The obtained Kappa score of 0.567 indicated substantial agreement among annotators, demonstrating that the annotation process was systematic, reproducible, and guided by well-defined labeling criteria.

The raw text was then preprocessed through the removal of special characters, URLs, and user mentions, among others, followed by tokenization and text normalization. Further, keywords were organized into categories reflecting common forms of hate expression observed in online discourse, such as *mshenzi* (uncivilized person), *shenzi* (barbaric), *malaya* (prostitute), *mbwa* (dog), *kumbafu* (idiot), and *mshirikina* (pagan), among others.

Feature extraction was subsequently performed using Term Frequency–Inverse Document Frequency (TF-IDF) and Word2Vec to obtain numerical vector representations. Text modeling was conducted iteratively using data partitioning into training (90,056), validation (11,258), and test (11,257) splits representing a ratio of 80:10:10, respectively. Benchmarking was done across traditional machine-learning models such as Naïve Bayes, Logistic Regression, SVM, Decision Tree, and deep-learning models such as RNN, GRU, BiLSTM, CNN, with final model selection guided by validation performance. Text inputs for the deep learning models were processed using TensorFlow's text vectorization layer. The vocabulary size was limited to 20,000 tokens, and each text instance was converted into a fixed-length sequence of 20 token indices. These sequences were then mapped into dense 128-dimensional vectors using a trainable embedding layer. Unlike static embedding methods, this trainable representation was learned jointly with the classifier, enabling the model to adapt to the linguistic patterns of Kiswahili hate-speech data. While this approach is computationally efficient and task-specific, it does not model contextual meaning as effectively as more recent transformer-based embeddings.

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

The image hate detection model was trained on a binary hate speech dataset labeled as hate or non-hate. Additionally, for the image modality, the qualitative strand similarly informed the image annotation guidelines to support consistent labeling of hate and not hate and further guided image screening and selection prior to model training. Image preprocessing was broken down into steps, such as resizing and normalization, to ensure compatibility with the deep learning architecture. The pre-trained weights of ResNet50V2 were used for initialization so that the model could benefit from its experience of large-scale feature learning from the ImageNet dataset. This drastically reduced computation and training time while enhancing the performance since the model is already good at recognizing general patterns in images. The pre-trained models were adapted for binary classification on hate and non-hateful content by discarding the last fully connected layers of this model, and they were replaced by custom layers such as Global Average Pooling (GAP) to reduce the spatial dimensions of feature maps into one feature vector in a computationally efficient manner. The fully connected classification head comprised a dense layer with 256 neurons and ReLU activation, followed by a dropout layer with a rate of 0.5 to reduce overfitting and improve model generalization. One neuron with a sigmoid activation function was used to output the probability of the image belonging to the hateful class. The model was fine-tuned on the collected dataset with the Adam optimizer, with a learning rate of 0.001 and beta values for adaptive learning. Binary cross-entropy loss was used to calculate the error in classification, and the batch size was kept at 32 to strike a balance between computational efficiency and learning stability. Further, the models were also trained for 50 epochs to achieve robustness. Early stopping based on validation loss was applied to avoid overfitting of the model. The earning rate scheduler decreased the learning rate dynamically in case of a plateau in validation performance. Eventually, to make the models robust and avoid overfitting, they were trained using data augmentation such as random horizontal flipping, rotation up to 15 degrees, zooming, and brightness adjustment. Data augmentation created training data variations, enabling the model to generalize to unseen samples better.

Hold-out validation was employed as part of the model validation process. This approach provided a more robust and generalizable estimate of model performance, particularly important in settings where dataset composition can influence results. In this method, the multimodal dataset was partitioned into 80:10:10 for the train, validation, and testing sets, respectively. To address class imbalance in the training data, the Synthetic Minority Oversampling Technique (SMOTE) was applied only to the training set, while the validation and test sets were retained as representations of real-world data. This way, we prevented the data leakage, which could have led to overfitting of the models. This increases confidence that the observed results reflect the model's true ability to generalize to unseen data.

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

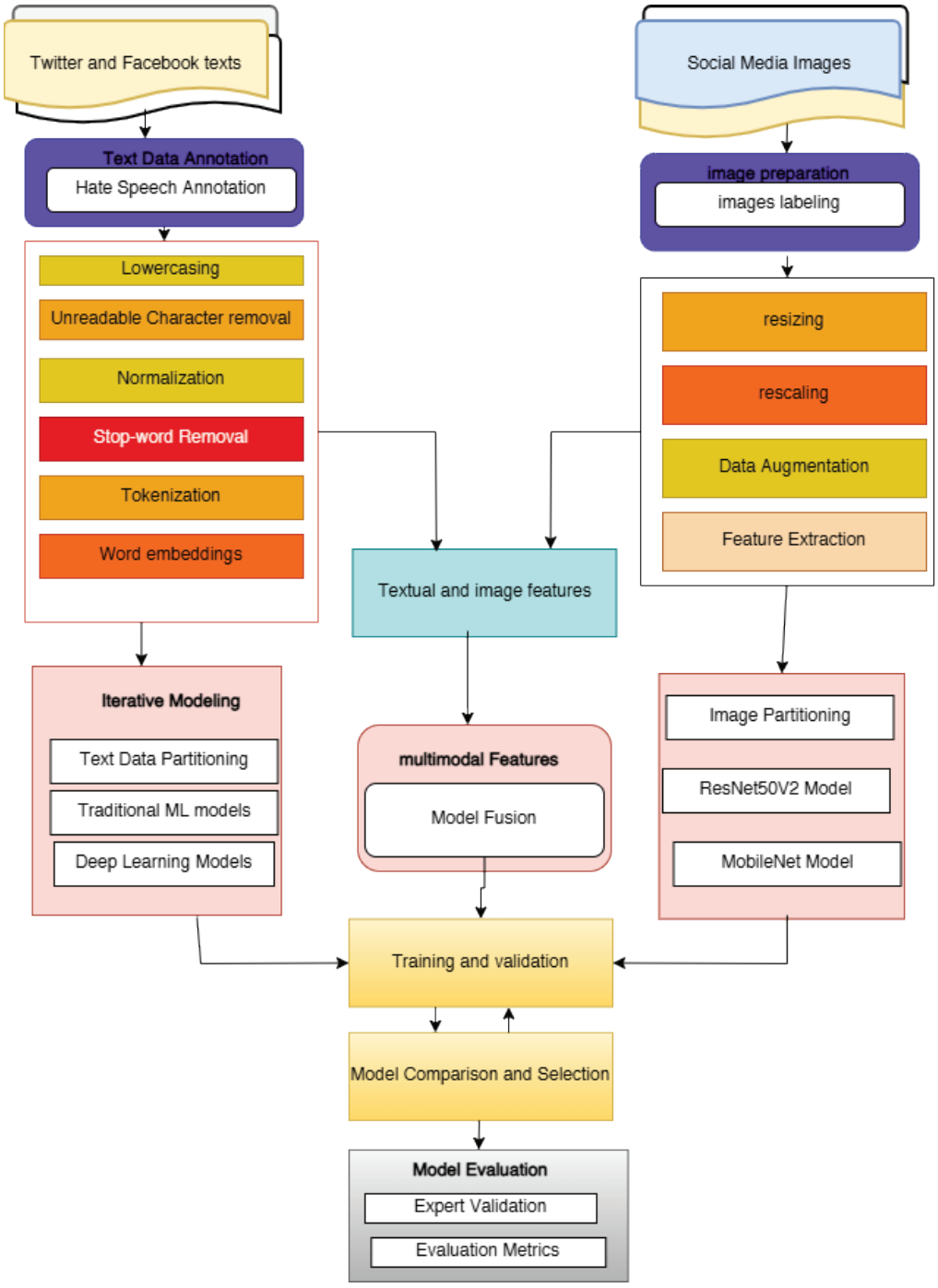


Figure 1: Multimodal architectural model

The multimodal hate speech detection model, ChujaHate, was implemented using a late fusion strategy since the image dataset with 2,607 samples was much smaller than the text dataset with 112,571 samples, creating a clear modality imbalance. The collection of hate speech image datasets was increasingly constrained by recent Application Programming Interfaces (APIs) and policy changes on Facebook and X (formerly Twitter), which limited the availability of multimodal data for this study, and further, image data required more extensive preprocessing than textual data. Given these differences in dataset size and modality characteristics, late fusion was considered more appropriate than early fusion. In late fusion, the probability outputs generated by the SoftMax layer of the BiLSTM model and the sigmoid probability outputs from the ResNet50V2 model were concatenated into a unified feature representation. This avoided the need for early alignment of heterogeneous features while still enabling effective multimodal integration [44]. Prior studies have shown

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

that late fusion could outperform early fusion across key performance metrics in multimodal settings, where textual and image embeddings are concatenated before classification [40]. The fused model was evaluated using metrics such as accuracy, precision, Recall, and F1-score, while hyperparameter tuning helps identify the best-performing configuration. Finally, a web-based user interface enables users to input text, images, or both and receive hate speech classification results in real time.

4. EXPERIMENTAL RESULTS

The experimental results of the unimodal and multimodal models developed for hate speech detection are presented in this section. The evaluation was conducted in two phases, where text and image models were trained independently on an annotated Kiswahili dataset and integrated at inference time. For the text modality, both traditional machine-learning baselines and deep learning model results are indicated in this section.

4.1. TRADITIONAL ML MODELS:

The comparative evaluation of traditional machine learning algorithms for Kiswahili hate speech detection, given in Table 2, reveals clear performance differences across models, both in baseline and optimized configurations.

The Multinomial Naïve Bayes model recorded the lowest performance across all metrics, with an accuracy of 78.50% and an F1-score of 77.46%. While its relatively higher precision of 79.37% suggests it can reasonably identify hate speech when predicted, the low Recall indicates that it fails to capture a significant portion of actual hate speech instances. This limitation can be attributed to its strong independence assumption, which is often unsuitable for complex linguistic patterns such as those found in Kiswahili and code-switched text. The Support Vector Machine (SVM) model emerged as one of the best-performing models, achieving an accuracy of 87.95%, a precision of 87.76%, a recall of 87.95%, and an F1 score of 87.63%. The balanced performance across all metrics indicates that SVM effectively handles both positive and negative classes. Its robustness can be linked to its ability to find optimal decision boundaries in high-dimensional feature spaces, especially when combined with TF-IDF features.

The Logistic Regression model demonstrated competitive performance, with an initial accuracy of 86.27% and an F1 score of 85.59%. However, after hyperparameter tuning, the model improved significantly, achieving an accuracy of 88.76% and an F1 score of 88.46%, surpassing the SVM model slightly. This improvement highlights the importance of parameter optimization, particularly in linear models where regularization plays a critical role in balancing bias and variance.

The Decision Tree model also performed well, with an accuracy of 87.99% and an F1 score of 87.87%. Its performance suggests that it is capable of capturing non-linear relationships in the data. However, decision trees are prone to overfitting, which may limit their generalization capability despite strong training performance.

The Random Forest model, an ensemble extension of decision trees, showed stable and slightly improved results compared to a single decision tree. The model achieved an accuracy of 87.63% and an F1-score of 84.04% after tuning. The marginal improvement during training of the random forest indicates that ensemble learning helped reduce overfitting and improved generalization, although the gains were not substantially higher than those of SVM or tuned Logistic Regression.

The results indicate that SVM and tuned Logistic Regression models provide the best balance between precision and Recall, making them highly suitable for hate speech detection tasks. While ensemble methods such as Random Forest offer robustness, their performance gains in this case are moderate. The findings also emphasize the importance of hyperparameter tuning, as seen in the significant improvement of Logistic Regression after optimization.

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

This finding is consistent with prior studies by researchers such as [9], who reported SVM as the top classical baseline with strong overall effectiveness and particularly strong hate-class separation, while [6] similarly demonstrated competitive SVM results on Kiswahili and English hate speech benchmarks.

Table 2: Traditional ML result comparison

ML ALGORITHM	ACCURACY	PRECISION	RECALL	F1-SCORE
Naïve Bayes	78.50	79.37	78.50	77.46
Support Vector Machine (SVM)	87.63	87.76	87.95	87.63
Logistic regression	88.76	88.51	88.76	88.46
Decision tree	87.99	87.87	87.99	87.87
Random Forest	87.63	87.75	87.64	84.84

4.2. DEEP LEARNING MODELS

i) LSTM Model Results

The Long Short-Term Memory (LSTM) model achieved strong performance, with an accuracy of 90.44%, demonstrating its ability to capture sequential dependencies in Kiswahili text. This confirms that recurrent architectures are well-suited for modeling linguistic structures in hate speech detection tasks. However, compared to traditional machine learning models such as SVM and LR, which were the best among classical approaches, the LSTM clearly surpasses them, highlighting the advantage of deep learning in capturing contextual relationships. Despite its strengths, the LSTM shows a slight imbalance between macro and weighted metrics, suggesting reduced effectiveness on minority classes. This indicates that while it learns general patterns well, it may miss subtle or less frequent hate expressions. Nevertheless, its performance is significantly higher than that of Logistic Regression, Decision Trees, and Random Forest models. The LSTM model, therefore, provides a strong foundation for deep learning-based text classification. It demonstrates that moving beyond feature-engineered approaches to sequence modeling leads to substantial gains in performance.

ii) GRU Model Results

The GRU model improves upon the LSTM by achieving an accuracy of 94.86%, indicating more efficient learning of contextual dependencies in the dataset. Compared to traditional machine learning models such as SVM, which already performed well, the GRU further demonstrates the superiority of deep learning approaches for this task. Its simpler architecture allows faster convergence while still maintaining high predictive power. However, the lower macro recall of 88.88% suggests that the model may not generalize equally well across all hate speech categories, particularly minority classes. Despite this, its weighted metrics remain very high, indicating strong overall classification capability. The GRU strikes a balance between computational efficiency and performance, making it practical for large-scale applications. It clearly outperforms classical approaches like Random Forest and Decision Trees. This reinforces the idea that neural architectures are better suited for capturing semantic and contextual nuances in Kiswahili text.

iii) Bidirectional Long Short-Term Memory Model (BiLSTM) Results

The Bidirectional LSTM (BiLSTM) achieved the best overall performance, with an accuracy of 97.33% and consistently high precision of 97.34%, Recall of 97.32%, and F1-score of 97.32%. This significantly surpasses both traditional machine learning models and other deep learning architectures such as LSTM and GRU. The superior performance of BiLSTM confirms that bidirectional context is critical for Kiswahili hate speech detection, where meaning often depends on both preceding and succeeding words. Unlike unidirectional models, BiLSTM captures richer semantic relationships, leading to improved classification accuracy across all classes. The close alignment between macro and weighted metrics further indicates

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

strong generalization and balanced learning. This model effectively minimizes both false positives and false negatives. Its dominance across all evaluation metrics demonstrates that it is the most suitable model for this task. Therefore, BiLSTM stands out as the most effective approach due to its ability to fully exploit contextual information.

iv) CNN Model Results

The CNN model achieved a strong accuracy of 93.42%, confirming its effectiveness in capturing local textual patterns such as key phrases associated with hate speech. Compared to traditional machine learning models like SVM, CNN still demonstrates superior performance, reinforcing the advantage of deep learning methods. However, when compared to recurrent architectures such as LSTM, GRU, and especially BiLSTM, its performance is slightly lower. This is mainly due to its limitation in capturing long-range dependencies in text. The relatively low macro recall 83.02% suggests that CNN struggles more with minority classes, focusing primarily on dominant patterns. Despite this, its high weighted scores indicate strong overall predictive ability. CNN remains computationally efficient and suitable for scenarios where speed is critical. While it does not outperform BiLSTM, it provides a solid alternative within deep learning approaches. Overall, it highlights the trade-off between efficiency and deep contextual understanding.

Among the traditional machine learning models, the Support Vector Machine (SVM) outperformed Logistic Regression when logistic regression is not fine-tuned, Decision Tree, and Random Forest, suggesting that margin-based classifiers are more effective for the curated hate speech dataset. Despite this strong performance, the deep learning models consistently achieved higher results than all traditional approaches. In particular, the Bidirectional Long Short-Term Memory (BiLSTM) model delivered the best overall performance, with an accuracy of 97.33%, precision of 97.34%, recall of 97.32%, and an F1-score of 97.32%. These consistently high metrics indicate that the model is both accurate and well-balanced in its predictions. The superior performance of the BiLSTM can be attributed to its ability to process text in both forward and backward directions, allowing it to capture richer contextual information. This is especially important in Kiswahili hate speech detection, where meaning often depends on surrounding words and sentence structure. As a result, the BiLSTM emerged as the most effective model for textual hate speech detection, making it more suitable for integration into a multimodal fusion model.

The summary showing the comparison of results is shown in Figure 2.

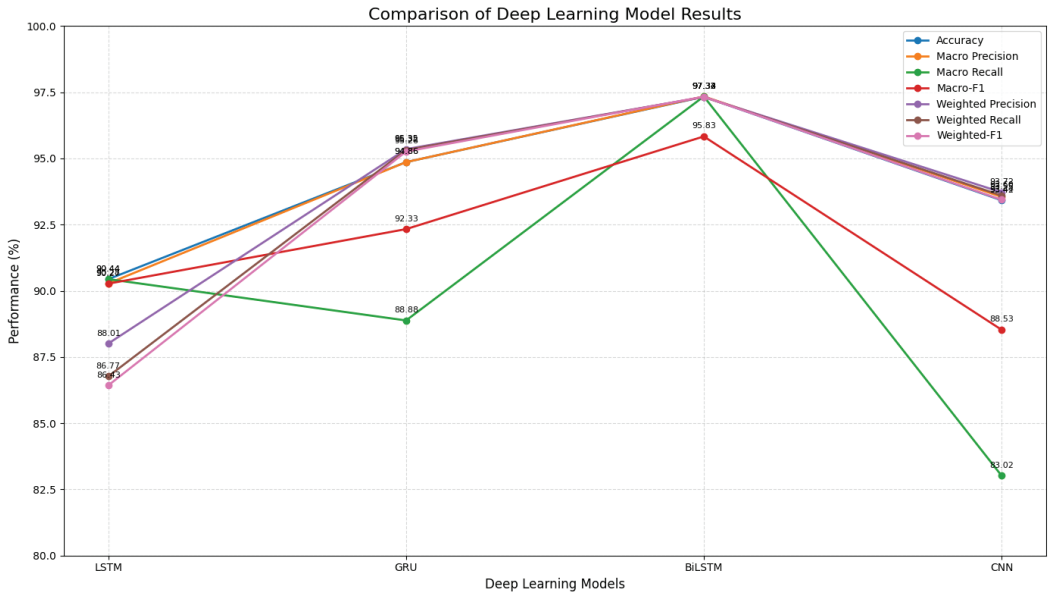


Figure 2: Deep Learning result comparison

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

4.3. CLASS-LEVEL EVALUATION (BILSTM)

Table 4 illustrates the class-wise performance of the BiLSTM model. The model achieved the highest F1-score in the Gender category with 98.43%, indicating very strong discrimination for the most represented class. The high performance was also observed for Not hate, Offensive, Sexual, Disability, Race, and Chronic disease, all of which recorded F1-scores above 95%. Compared to a lower performance was observed for Religion (92.16%) and Tribe (93.56%), suggesting that these categories were relatively more difficult to distinguish. This may be attributed to contextual overlap with other abusive or identity-based expressions, fewer representative training examples, or semantic ambiguity in the dataset. The overall class-level results confirm that BiLSTM maintained high and relatively balanced performance across the nine hate-speech categories.

Table 3: BiLSTM model Class-level evaluation

Class	Precision (%)	Recall (%)	F1-score (%)	Support (%)
Not hate	96.87	96.56	96.71	320
Offensive	98.90	94.50	96.65	382
Sexual	98.22	93.80	95.96	823
Gender	97.89	98.98	98.43	6455
Disability	97.45	95.89	96.66	2508
Race	98.88	93.62	96.17	94
Chronic disease	94.74	97.67	96.18	129
Religion	88.92	95.64	92.16	344
Tribe	94.03	93.10	93.56	203

4.4. IMAGE-BASED MODELS' PERFORMANCE

The image classification pipeline based on the ResNet50V2 architecture achieved an overall accuracy of 89.77%, outperforming MobileNetV2 with an accuracy of 75.67. This demonstrates the ResNet50V2 model's ability to identify visual indicators of hate speech within social media images. Although this performance is slightly lower than that of the text-based BiLSTM model, the results remain significant given the inherent complexity of visual data and the contextual ambiguity often associated with images. Unlike text, where meaning is explicitly encoded in language, images frequently require contextual interpretation, making hate-related cues more subtle and sometimes difficult to generalize.

The model effectively learned to recognize key visual patterns commonly associated with hateful content, including offensive symbols, provocative imagery, and contextually charged visuals. The confusion matrix analysis indicated a high number of correctly classified instances, particularly for clearly distinguishable hate-related images. This suggests that the ResNet50V2 model was successful in extracting meaningful spatial features and leveraging pre-trained representations to identify patterns relevant to the classification task.

However, some misclassifications were observed, particularly in cases where images lacked clear semantic indicators or required external context for interpretation. False Positives occurred when non-hateful images contained visually similar elements to known hate symbols, leading the model to misinterpret them. Conversely, False Negatives were noted in instances where hate speech was implied through subtle or culturally specific imagery that the model could not fully capture. These limitations highlight a key challenge in image-based hate speech detection, where meaning is often implicit and dependent on contextual or cultural knowledge beyond pixel-level features.

Despite these challenges, the ResNet50V2 model demonstrated robustness and strong

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

generalization capability, benefiting from transfer learning and deep feature extraction. Its performance confirms that visual data provides valuable complementary information in hate speech detection, even though it may not be sufficient when used in isolation. These findings underscore the importance of integrating image analysis with textual understanding to improve overall detection accuracy. The comparison of the results is summarized in Table 5.

Table 4: MobileNetV2 and ResNet50V2 Models Performance"

MODELS	ACCURACY(%)	PRECISION(%)	RECALL(%)	F1(%)
MobileNetV2	75.67	78.95	75.67	73.05
ResNet50V2	89.77	89.76	89.76	89.76

4.5. MULTIMODAL FUSION RESULTS

The multimodal fusion framework adopted a decision-level late fusion strategy. In this approach, the BiLSTM text classifier and the ResNet50V2 image classifier were first trained independently on their respective modalities. During inference, the probability outputs generated by the SoftMax layer of the BiLSTM model and the sigmoid probability outputs from the ResNet50V2 model were concatenated into a unified feature representation. The combined prediction vectors were then passed through a fully connected neural fusion layer followed by SoftMax activation to generate the final hate speech classification. This late fusion approach was selected because the textual and image datasets differed substantially in size and structure, making direct feature-level alignment less suitable for the current study. The approach also reduced the complexity associated with heterogeneous multimodal feature synchronization while still enabling complementary contextual learning across modalities.

The performance of the multimodal model, as illustrated in Figure 3, demonstrates strong and balanced results across all evaluation metrics. The model achieved an accuracy of 92.15%, a precision of 91.87%, a recall of 92.35%, and an F1-score of 92.11%. The close alignment of these metrics indicates that the model maintains a good balance between correctly identifying hate speech and minimizing false classifications. In particular, the slightly higher Recall suggests that the model is highly effective in capturing hate speech instances, reducing the likelihood of missed detections.

The multimodal approach significantly reduced misclassification rates by leveraging the strengths of both text and image models. For instance, it addressed challenges such as sarcasm or implicit hate that may not be easily detected in text-only models, as well as contextual nuances in memes that image-only models might fail to interpret. By combining these complementary perspectives, the system achieved more robust and reliable predictions.

Furthermore, the fusion of the best-performing text model (BiLSTM) and image model (ResNet50V2) resulted in superior performance compared to individual unimodal approaches. While the BiLSTM alone achieved higher standalone accuracy, the multimodal model provided a more balanced and generalized performance across different types of data. This demonstrates that integrating textual and visual features enhances the overall effectiveness of Kiswahili hate speech detection systems. From the results, we confirm that multimodal fusion is a valuable strategy for improving classification performance in complex, real-world scenarios.

Although transformer-based models such as SwahBERT are widely regarded as state-of-the-art in NLP tasks, the results obtained from the BiLSTM model in this study demonstrate that competitive performance can still be achieved using less computationally intensive architectures. This is particularly important in real-world deployment scenarios where computational efficiency and latency are critical factors. The findings suggest that while transformer-based models such as SwahBERT may provide superior contextual

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

representations, the performance gap may not always justify the significantly higher computational cost, especially in applications targeting resource-constrained environments. The BiLSTM model, when properly tuned and combined with effective preprocessing techniques, provides a viable alternative that balances performance and efficiency. Furthermore, the integration of ResNet50v2 for image classification within the multimodal framework enhances the overall system's capability to detect hate speech beyond text alone. This multimodal approach represents a key contribution of the study, distinguishing it from purely text-based transformer models. Based on the multimodal approach, we developed an interactive hate speech detection interface called *ChujaHate*, which was validated by experts, presenting another key contribution of the study.

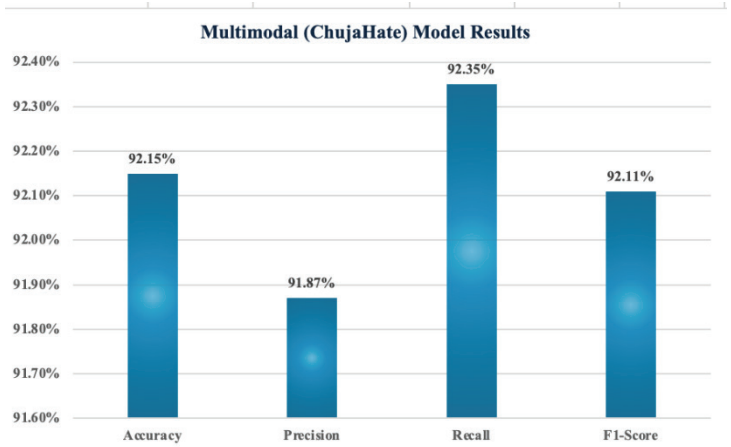


Figure 3: Multimodal Hate Speech Model results

4.6. COMPARATIVE ANALYSIS BETWEEN THE REVIEWED AND THE DEVELOPED MULTIMODAL FUSION MODEL

The findings of this study demonstrate that multimodal hate speech detection in low-resource African languages can achieve performance levels comparable to, and in some cases exceeding, those reported in prior multilingual hate speech studies. While previous studies, such as [1], achieved strong results using transformer-based architectures on code-switched English-Kiswahili text, the current study extends this body of knowledge by integrating both textual and visual modalities within a unified multimodal framework. Unlike earlier approaches that relied solely on textual analysis, the ChujaHate framework demonstrated that combining BiLSTM and ResNet50V2 models significantly improves contextual understanding, particularly for implicit and image-supported hate speech. This finding aligns with studies by [27] and [45], which observed that multimodal fusion provides richer semantic representation than unimodal systems.

The study further contributes to low-resource language research by demonstrating that computationally efficient architectures such as BiLSTM can still achieve highly competitive performance despite the increasing dominance of transformer-based models.

Although transformer-based architectures such as mBERT, AfroXLMR, CLIP, and multimodal vision-language transformers have demonstrated strong performance in multilingual hate speech detection tasks, they were not implemented in the current study due to computational and dataset-related constraints. Training and fine-tuning such models require substantially larger multimodal datasets, higher GPU memory requirements, and longer training durations. The present study, therefore, prioritized computationally efficient architectures, including BiLSTM and ResNet50V2, which remain suitable for deployment in resource-constrained African NLP environments. Nevertheless, transformer-based multimodal models remain an important direction for future research.

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

5. ETHICS STATEMENT

Data collection adhered to social media platform guidelines, ensuring compliance with privacy policies. User data was anonymized to protect personal information, and only publicly available datasets were used. Further, this study was undertaken under a data collection license issued to the researcher by the Kenya National Commission for Science, Technology, and Innovation (NACOSTI). Ethical approval was also obtained from the United States International University - Africa prior to the commencement of the study.

6. LIMITATIONS

The text dataset was substantially larger than the image dataset, creating a modality imbalance that may have influenced multimodal learning and fusion performance. The hate speech dataset may also not have fully captured all forms of hate speech, particularly sarcasm, coded Kiswahili expressions, and culturally specific references.

Although the study evaluated the performance of text-only, image-only, and multimodal models, a comprehensive ablation analysis involving multiple fusion strategies and feature contribution experiments was not conducted. Specifically, the study did not compare early fusion, attention-based fusion, and transformer-based multimodal architectures due to computational limitations and the relatively small size of the image dataset. Future work should therefore include expansion of the dataset to ensure balance, extensive ablation experiments to better quantify modality contribution, feature interaction, and the effectiveness of alternative multimodal fusion strategies.

7. RECOMMENDATIONS AND FUTURE WORK

i) Community-Driven Annotation and Local Collaboration:

Establishing community-based annotation hubs through partnerships with universities, research institutions, and civic technology organizations would also provide a sustainable approach to developing high-quality datasets for low-resource languages such as Kiswahili.

ii) Expansion to Additional Modalities:

Future research should extend beyond text and image analysis by incorporating additional modalities such as audio, video, emojis, and other forms of digital expression commonly used on platforms like TikTok and WhatsApp. Hate speech communication on modern social media platforms is increasingly multimodal, and integrating these additional data forms would provide a more comprehensive understanding of how harmful content is created, shared, and interpreted online.

iii) Institutional Integration and Policy Alignment:

Government institutions, digital rights organizations, and social media platforms should work collaboratively to integrate multimodal hate speech detection systems into existing policy enforcement and content moderation frameworks. The adoption of AI-assisted moderation tools can support faster and more proactive detection of harmful online content while reducing the workload placed on human moderators.

iv) Public Awareness and Digital Literacy:

There is also a need to strengthen public awareness on the role of artificial intelligence and machine learning in combating online hate speech. Governments, civil society organizations, educational institutions, and private sector stakeholders should develop digital literacy programs that educate users on how automated hate speech detection systems operate, their benefits, and their limitations. Such initiatives would promote responsible online

http://dx.doi.org/10.21622/ACE.2026.06.1.2076

behavior, encourage informed use of digital platforms, and foster a culture of respectful and inclusive digital communication.

5. CONCLUSION

This study introduced *ChujaHate*, which, to the best of our knowledge, is the first multimodal hate speech detection framework developed specifically for Kiswahili social media content. The study addressed an important gap in low-resource African language NLP research by integrating both textual and visual modalities in hate speech detection. The findings showed that the text-based model achieved the strongest overall performance across the nine fine-grained hate speech categories, while the image-based model also demonstrated promising capability in detecting hateful visual content. When both modalities were combined using a late fusion approach, the multimodal framework showed complementary benefits, confirming that hateful meaning in social media is often distributed across both text and images. However, the improvement achieved through multimodal fusion was moderated by the relatively smaller image dataset, which created a modality imbalance during model training.

Beyond the technical findings, the study has practical significance for content moderation in Kiswahili-speaking online communities, where hate speech is frequently communicated through culturally contextualized combinations of language, memes, and imagery. By incorporating both text and image information, *ChujaHate* provides a more context-aware and inclusive framework that can support social media platforms, policymakers, researchers, and civil society organizations in developing more effective moderation systems for low-resource multilingual environments. The study therefore contributes not only to the advancement of multimodal machine learning research but also to ongoing efforts aimed at promoting safer and more responsible digital communication spaces in Africa.

Despite these contributions, the study faced several limitations, including the relatively small image dataset, possible subjectivity and cultural bias during manual annotation, and the use of non-contextual embeddings in parts of the modeling pipeline, which may have limited the capture of nuanced contextual meanings in Kiswahili hate speech.

Nevertheless, the findings of this study demonstrate that multimodal machine learning can provide more context-aware hate speech detection than unimodal systems, particularly in low-resource multilingual environments where hateful meaning is often distributed across both text and imagery. While the text-only BiLSTM achieved the highest standalone accuracy, the multimodal framework improved contextual robustness and practical applicability in real-world social media scenarios. The study, therefore, highlights the importance of balancing predictive performance, computational efficiency, and deployment feasibility when designing AI systems for African language technologies.

Future research should therefore focus on developing larger and more balanced multimodal datasets, adopting transformer-based models such as mBERT fine-tuned for Kiswahili, exploring advanced fusion techniques such as attention-based and cross-modal learning, and extending analysis to audio and video modalities to further improve multimodal hate speech detection systems in African languages.

REFERENCES

[1] F. N. Njung'e, A. M. Oirere, and R. N. Ndung'u, "A Comparative Study of Transformer-based Models for Hate-Speech Detection in English-Kiswahili Code-Switched Social Media Text," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 13, no. 5, pp. 181–186, Oct. 2024, doi: https://doi.org/10.30534/ijatcse/2024/011352024.

[2] G. Arya *et al.*, "Multimodal Hate Speech Detection in Memes using Contrastive

Language-Image Pre-training," *IEEE access*, vol. 12, pp. 22359–22375, Jan. 2024, doi: https://doi.org/10.1109/access.2024.3361322.

[3] A. Marshan, F. Nasreen, A. Ioannou, and K. Spanaki, "Comparing Machine Learning and Deep Learning Techniques for Text Analytics: Detecting the Severity of Hate Comments Online," *Information Systems Frontiers*, pp. 1–19, Nov. 2023, doi: https://doi.org/10.1007/s10796-023-10446-x.

[4] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, "Hate speech detection: Challenges and solutions," *PLOS ONE*, vol. 14, no. 8, p. e0221152, Aug. 2019, doi: https://doi.org/10.1371/journal.pone.0221152.

[5] United Nations, "United Nations Strategy and Plan of Action on Hate Speech," *United Nations Digital Library System*, 2019. https://digitallibrary.un.org/record/3889290

[6] N. Onyango, L. Wanzare, and J. I. Obuhuma, "Swahili and Code-Switched English-Swahili Political Hate Speech Detection Textual Dataset," *Data Intelligence*, vol. 7, no. 3, pp. 819–850, 2025, doi: https://doi.org/10.3724/2096-7004.di.2025.0053.

[7] B. C. Solomon, Matthew, A. Hemmen, and J. N. Druckman, "Illusory interparty disagreement: Partisans agree on what hate speech to censor but do not know it," *Proceedings of the National Academy of Sciences*, vol. 121, no. 39, Sep. 2024, doi: https://doi.org/10.1073/pnas.2402428121.

[8] F. M. Ndahinda and A. S. Mugabe, "Streaming Hate: Exploring the Harm of Anti-Banyamulenge and Anti-Tutsi Hate Speech on Congolese Social Media," *Journal of Genocide Research*, vol. 26, no. 1, pp. 1–25, May 2022, doi: https://doi.org/10.1080/14623528.2022.2078578.

[9] E. Ombui, M. Karani, and L. Muchemi, "Annotation Framework for Hate Speech Identification in Tweets: Case Study of Tweets During Kenyan Elections," *IEEE Xplore*, May 01, 2019. https://ieeexplore.ieee.org/document/8764868 (accessed Apr. 14, 2022).

[10] T. M. Ababu and M. M. Woldeyohannis, "Afaan Oromo Hate Speech Detection and Classification on Social Media," in *ACL Anthology*, Proc. 13th Language Resources and Evaluation Conf, Jun. 2022, pp. 6612–6619.

[11] O. Oriola and Kotze E., "Improved semi-supervised learning technique for automatic detection of South African abusive language on Twitter," *South African Computer Journal*, vol. 32, no. 2, pp. 56–79, Dec. 2020, doi: https://doi.org/10.18489/sacj.v32i2.847.

[12] S. Badillo *et al.*, "An Introduction to Machine Learning," *Clinical Pharmacology & Therapeutics*, vol. 107, no. 4, pp. 871–885, Mar. 2020, doi: https://doi.org/10.1002/cpt.1796.

[13] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, no. 1, pp. 512–515, May 2017, doi: https://doi.org/10.1609/icwsm.v11i1.14955.

[14] M. Wiegand, J. Ruppenhofer, Anna Marie Schmidt, and C. Greenberg, "Inducing a Lexicon of Abusive Words – a Feature-Based Approach," in *Publication Server of the Institute for German Language (Institute for German Language)*, Leibniz Institute for the German Language: Proc. NAACL-HLT, Jan. 2018. doi: https://doi.org/10.18653/v1/n18-1095.

[15] P. Mishra, M. D. Tredici, H. Yannakoudakis, and E. Shutova, "Author Profiling for Hate Speech Detection," 2019, doi: https://doi.org/10.48550/arXiv.1902.06734.

[16] G. Martin, M. E. Mswahili, Y.-S. Jeong, and J. Woo, "SwahBERT: Language Model of

Swahili," in *ACLWeb* , Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 303–313. doi: https://doi.org/10.18653/v1/2022.naacl-main.23.

[17] O. F. Babu, M. Jahan, A. Faisal, Md. S. Islam, and R. Khan, "Bangla Hate Speech Detection System Using Transformer-Based NLP and Deep Learning Techniques," in *Proc. 3rd Asian Conf. Innovation in Technology (ASIANCON)*, Aug. 2023, pp. 1–6. doi: https://doi.org/10.1109/asiancon58793.2023.10269919.

[18] A. S. Alammary, "BERT Models for Arabic Text Classification: A Systematic Review," *Applied Sciences*, vol. 12, no. 11, p. 5720, Jun. 2022, doi: https://doi.org/10.3390/app12115720.

[19] D. Adelani, G. Neubig, S. Ruder, and S. Rijhwani, "MasakhaNER 2.0: Africa-centric Transfer Learning for Named Entity Recognition," *arXiv preprint arXiv:2210.12391*, 2022, doi: https://doi.org/10.48550/arXiv.2210.12391.

[20] A. Kaliba, "Performance Assessment of a New Swahili Lexicon (SWAHILILex.01) Tagged by Native Speakers for Polarity Analysis," May 2023, doi: https://doi.org/10.36227/techrxiv.22806782.

[21] M. Kandagor, "THE PLACE OF KISWAHILI IN THE TWENTY-FIRST CENTURY | Mark M. Kandagor," in *Mu.ac.ke*, Youth, Globalization, and Society in Africa and Its Diaspora, 2020, p. 114.

[22] C. Jacobs, N. C. Rakotonirina, E. A. Chimoto, B. A. Bassett, and H. Kamper, "Towards hate speech detection in low-resource languages: Comparing ASR to acoustic word embeddings on Wolof and Swahili," *arXiv.org*, 2023. https://arxiv.org/abs/2306.00410

[23] A. Irfan, D. Azeem, S. Narejo, and N. Kumar, "Multi-Modal Hate Speech Recognition Through Machine Learning," *Proc. IEEE 1st Karachi Section Humanitarian Technology Conf. (KHI-HTC)*, pp. 1–6, Jan. 2024, doi: https://doi.org/10.1109/khi-htc60760.2024.10482031.

[24] A. G. Debele and M. M. Woldeyohannis, "Multimodal Amharic Hate Speech Detection Using Deep Learning," *Proc. Int. Conf. Information and Communication Technology for Development for Africa (ICT4DA)*, pp. 102–107, Nov. 2022, doi: https://doi.org/10.1109/ict4da56482.2022.9971436.

[25] K. Perifanos and D. Goutsos, "Multimodal Hate Speech Detection in Greek Social Media," *Multimodal Technologies and Interaction*, vol. 5, no. 7, p. 34, Jun. 2021, doi: https://doi.org/10.3390/mti5070034.

[26] P. Kapil and A. Ekbal, "A transformer based multi task learning approach to multimodal hate detection," *Natural Language Processing Journal*, vol. 11, p. 100133, Feb. 2025, doi: https://doi.org/10.1016/j.nlp.2025.100133.

[27] M. D. Belete and Girma Kassa Alitasb, "Identification of Hateful Amharic Language Memes on Facebook using Deep Learning Algorithms," *Systems and Soft Computing*, vol. 7, pp. 200258–200258, Apr. 2025, doi: https://doi.org/10.1016/j.sasc.2025.200258.

[28] F. K. Saddozai, S. K. Badri, D. Alghazzawi, A. Khattak, and M. Z. Asghar, "Multimodal hate speech detection: a novel deep learning framework for multilingual text and images," *PeerJ Computer Science*, vol. 11, p. e2801, Apr. 2025, doi: https://doi.org/10.7717/peerj-cs.2801.

[29] F. Vargas *et al.*, "HausaHate: An Expert Annotated Corpus for Hausa Hate Speech Detection," *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH 2024)*, pp. 52–58, 2024, doi: https://doi.org/10.18653/v1/2024.woah-1.5.

[30] B. Vidgen and T. Yasseri, "Detecting weak and strong Islamophobic hate speech on social media," *Journal of Information Technology & Politics*, vol. 17, no. 1, pp. 66–78,

Dec. 2019, doi: https://doi.org/10.1080/19331681.2019.1702607.

[31] A. K. Diallo and K. Abainia, "Offensive Language Detection in Code-Mixed Bambara-French Corpus: Evaluating machine learning and deep learning classifiers," *2023 International Conference on Decision Aid Sciences and Applications (DASA)*, pp. 121–125, Sep. 2023, doi: https://doi.org/10.1109/dasa59624.2023.10286577.

[32] S. H. Muhammad *et al.*, "AfriSenti: A Twitter Sentiment Analysis Benchmark for African Languages," Feb. 2023, doi: https://doi.org/10.48550/arxiv.2302.08956.

[33] F. Ieracitano, C. Balenzano, S. Girardi, C. G. Gemmano, and F. Comunello, "Online Hate Speech as a Moral Issue: Exploring Moral Reasoning of Young Italian Users on Social Network Sites," *Social Science Computer Review*, vol. 42, no. 1, p. 089443932311611, Mar. 2023, doi: https://doi.org/10.1177/08944393231161124.

[34] R. A. Akbar, A. Mulyana, and M. Amalia, "Legal Challenges In The Age Of Social Media: Protecting Citizens From Misuse Of Information," *Golden Ratio of Law and Social Policy Review*, vol. 3, no. 1, pp. 14–25, Dec. 2023, doi: https://doi.org/10.52970/grlspr.v3i1.328.

[35] E. Ombui, L. Muchemi, and P. Wagacha, "Psychosocial Features for Hate Speech Detection in Code-switched Texts," *International Journal of Information Technology and Computer Science*, vol. 13, no. 6, pp. 29–47, Dec. 2021, doi: https://doi.org/10.5815/ijitcs.2021.06.03.

[36] A. Chhabra and D. K. Vishwakarma, "A literature survey on multimodal and multilingual automatic hate speech identification," *Multimedia Systems*, vol. 29, no. 3, Jan. 2023, doi: https://doi.org/10.1007/s00530-023-01051-8.

[37] F. Wu, G. Chen, J. Cao, Y. Yan, and Z. Li, "Multimodal Hateful Meme Classification Based on Transfer Learning and a Cross-Mask Mechanism," *Electronics*, vol. 13, no. 14, p. 2780, Jul. 2024, doi: https://doi.org/10.3390/electronics13142780.

[38] F. Chen, X. Li, Z. Li, C. Zhou, and J. Sheng, "Multimodal Rumor Detection via Multimodal Prompt Learning," *2022 International Joint Conference on Neural Networks (IJCNN)*, vol. 7, pp. 1–8, Jun. 2024, doi: https://doi.org/10.1109/ijcnn60899.2024.10650974.

[39] P. Kapil and A. Ekbal, "A transformer based multi task learning approach to multimodal hate detection," *Natural Language Processing Journal*, vol. 11, p. 100133, Feb. 2025, doi: https://doi.org/10.1016/j.nlp.2025.100133.

[40] P. Vijayaraghavan, H. Larochelle, and D. Roy, "Interpretable Multi-Modal Hate Speech Detection," *arXiv.org*, 2021. https://arxiv.org/abs/2103.01616? (accessed Jun. 06, 2026).

[41] J. L. Imbwaga, N. B. Chittaragi, and S. G. Koolagudi, "Automatic hate speech detection in audio using machine learning algorithms," *International Journal of Speech Technology*, vol. 27, no. 2, pp. 447–469, Jun. 2024, doi: https://doi.org/10.1007/s10772-024-10116-6.

[42] Z. Zhao, Z. Zhang, and F. Hopfgartner, "Detecting Toxic Content Online and the Effect of Training Data on Classification Performance," *EasyChair Preprints*, Apr. 2019, doi: https://doi.org/10.29007/z5xk.

[43] M. Zampieri, S. Rosenthal, Preslav Nakov, Alphaeus Dmonte, and T. Ranasinghe, "OffensEval 2023: Offensive language identification in the age of Large Language Models," *Natural language engineering*, vol. 29, no. 6, pp. 1416–1435, Nov. 2023, doi: https://doi.org/10.1017/s1351324923000517.

http://dx.doi.org/10.21622/ACE.2026.06.1.1955

A REMOTE CONTROLLED IOT SMART METER MONITORING SYSTEM WITH THEFT DETECTION

Ayoola E. Akindele[1], Ivan Kholodilin[2] and Afolabi Awodeyi[3]

[1,2] South Ural State University, Department of Mechatronics and Robotics Engineering, Chelyabinsk, Russia

[3] Southern Delta University, Department of Computer Engineering, Ozoro, Delta State, Nigeria

Emails: {ayoola.akindele@ieee.org, kholodilinii@susu.ru, awodeyiai@dsust.edu.ng}

Received 12 February 2026 - Accepted 21 March 2026 - Published 10 June 2026

ABSTRACT

This research addresses the challenges of energy administration and security in developing nations. It proposes an Internet of Things (IoT)-based smart meter monitoring system that integrates security controls to prevent theft and allows for remote monitoring of energy usage. The system utilizes an Atmega microcontroller for connectivity, a GSM module for sending data, a non-invasive AC current sensor for accurate readings, and a user interface for data visualization and control. Upon detecting theft, the system sends an alert to the user and disconnects the power supply. Users can also remotely monitor energy consumption and turn off the meter through the GSM connection.

Keywords: *ESmart Meter; Energy Monitoring; Internet of things; Atmega Microcontroller; Sensing system; Current Sensor; GSM interaction*

1. INTRODUCTION

Technology is reshaping our world, and the Internet of Things (IoT) is at the heart of this change. IoT connects everyday devices from smart watches to traffic lights, allowing them to share data effortlessly. This isn't just convenience; it's a revolution. In healthcare, wearables monitor vitals in real time, helping doctors catch issues early. Smart cities use IoT to ease traffic, cut energy waste, and even manage trash collection efficiently. Meanwhile, military systems leverage IoT for better situational awareness, and retailers use it to personalize shopping and track inventory. Factories automate processes, and oil pipelines detect leaks, all thanks to IoT. One exciting IoT innovation is the smart energy meter. Beyond tracking usage, it enhances security by detecting tampering, lets users monitor consumption remotely, and even allows control of appliances via a simple app. This isn't just about saving money; it's about smarter, safer energy use. By merging IoT with energy solutions, we're stepping into a future where sustainability and efficiency go hand in hand. The potential is huge, and the best part? It's just the beginning.

2. REVIEW OF RECENT SMART METER AND ELECTRICITY THEFT DETECTION SYSTEMS

Recent research on IoT-based smart energy meters has focused on improving energy efficiency, billing accuracy, and theft detection. For example, the research by Yaghmaee and Hejazi in 2018[1] demonstrated an IoT-based metering system for real-time monitoring, but it lacked integrated theft prevention mechanisms. Also, Barman et al. in 2018 [8] proposed a smart meter for efficient energy utilization, but their system did not incorporate automated disconnection upon detecting anomalies. Tahir et al. in 2022 [3] introduced blockchain-based smart metering to enhance data security, although the approach increases system

http://dx.doi.org/10.21622/ACE.2026.06.1.1955

complexity and cost. In contrast, simpler GSM-based systems, such as the research of Agnihotri in 2022 [9], focus primarily on communication without advanced detection mechanisms. More recent studies have explored electricity theft detection using IoT and machine learning techniques, but many of these systems rely on complex data-driven models that require extensive datasets and computational resources, which limit their applicability in resource-constrained environments.

When comparing, the proposed system adopts a hardware-driven detection approach using tamper sensors and current monitoring that offers lower computational requirements, faster response time, and ease of implementation in real-world environments.

3. SYSTEM ARCHITECTURE AND THEFT DETECTION ALGORITHM

3.1. SYSTEM ARCHITECTURE

The proposed system follows a modular architecture consisting of four primary layers:

a) Sensing Layer

Includes the AC current sensor and tamper switch for detecting energy consumption and unauthorized access.

b) Processing Layer

The ATmega328p microcontroller processes sensor inputs and executes decision-making algorithms.

c) Communication Layer

i) ESP8266 module for cloud-based monitoring
ii)GSM module for SMS alerts and remote control

d) Control Layer

This is the relay module responsible for disconnecting power upon theft detection.

3.2. THEFT DETECTION ALGORITHM

The theft detection mechanism is based on threshold monitoring and tamper event detection. The algorithm's operation is as follows:

a) Initialize system parameters and communication modules

b) Continuously read current sensor values

c) Monitor tamper switch status

d) Compare the measured current with the expected consumption pattern

e) If the tamper signal is triggered OR abnormal current deviation is detected:

i) Flag as theft event
ii)Send SMS alert via GSM module
iii) Upload event to the cloud via Wi-Fi
iv) Activate the relay to disconnect the supply

f) Else:

i) Continue normal monitoring

g) Repeat the process in a real-time loop

http://dx.doi.org/10.21622/ACE.2026.06.1.1955

4. SYSTEM ANALYSIS AND DESIGN

This section delves into the intricacies of the research, encompassing the technological and systemic approaches employed. It details the justifications for the chosen methodologies and components, highlighting their advantages and suitability in achieving the research objectives. The section proceeds with an extensive description of:

a) Hardware Design and Algorithms: This includes detailed explanations of the hardware elements, associated algorithms, computations performed, and development environments used to manage the device.

b) Software Development: This explores the programming languages, frameworks, and functionalities employed in the developed software.

c) Sequencing Methods: This explains the essential sequencing methods implemented to ensure proper functioning and interaction of all system components.

SYSTEM ANALYSIS

The design and implementation of the IoT Smart Energy Meter Monitoring and Energy Theft Sensing System comprises two key sections:

a) Hardware Components: This section focuses on the selection, integration, and functionalities of the various hardware components utilized in the system. Prior to integration with the ATmega microcontroller, a thorough analysis of each individual hardware component is conducted, elucidating its role and functionalities within the system.

b) Software Development: This section explores the software specifically developed for the system. It delves into the programming languages, frameworks, and functionalities employed to achieve the desired outcomes.

Table 1, summarizing the primary elements involved in the research project, is included within this section. This table provides a concise overview of the essential hardware and software components utilized in the system design and implementation.

Table 1: Primary elements of this research

COMPONENT No.	System component UNIT	PARTS AND DEPICTIONS
1	Power Unit	i. 12V DC lithium-ion battery ii. DC-DC Multi-output Buck Converter (3.3V/5V/9V/12V) iii. A step-down transformer 240/12V 50Hz iv. Optocoupler PC817 IC
2	Sensing & Control Unit	i. Tamper sensor ii. AC Current Sensor iii. ESP8266 Transmitting Module iv. Input voltage of 12V v. ATMega328p microcontroller vi. GSM Module vii. 12V DC supply (Battery) viii. 5V relay
3	Communication Unit	i. ESP8266 Transmitting Module GSM Module
4	Software	i. Arduino IDE (C++)

The system comprises several interconnected components fulfilling specific functions:

i) Sensing:

Analog Sensors: The GSM module and tamper switch are connected to specific analog pins of

http://dx.doi.org/10.21622/ACE.2026.06.1.1955

the microcontroller, enabling them to provide analog outputs between 0 and 1023, readable by the system.

ii) Communication:

Digital Pins: The ESP8266 module, offering Wi-Fi connectivity, is connected to designated digital pins of the microcontroller.

SPI Bus: The Arduino communicates with the Wi-Fi shield utilizing the SPI bus.

iii) Power Control:

Relay: Controlled by a HIGH signal through the tamper switch and switched off by a LOW signal, the relay connects loads to specific digital pins of the Arduino Uno. This enables monitoring and theft detection functionalities upon activation.

A visual representation of these connections is depicted in Figure 1

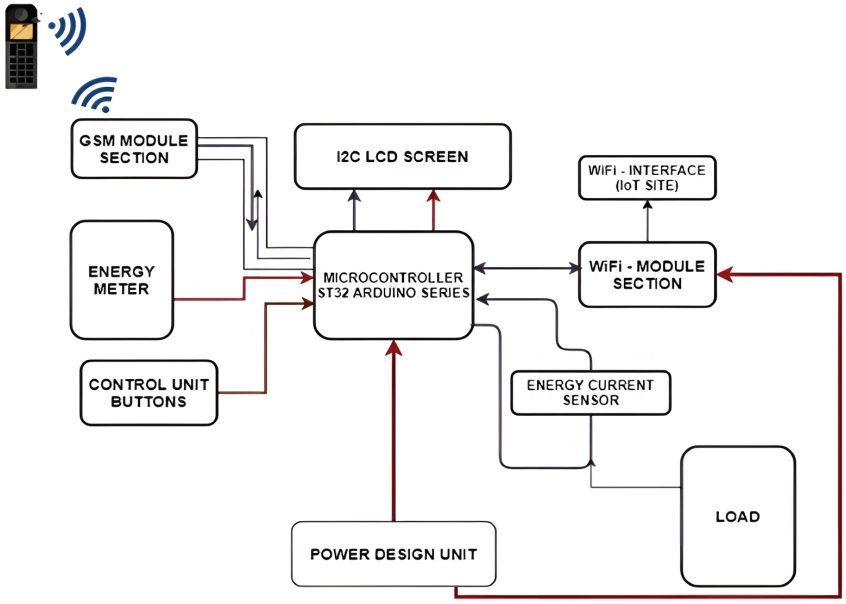


Figure 1: Hardware Block Diagram

HARDWARE DESIGN

This section looks into the essential hardware components of the IoT Smart Energy Meter Monitoring and Energy Theft Sensing System, detailing their design and implementation. It focuses on the circuit design and interconnections between components crucial for achieving the system's functionality. These components are described below:

Tamper Switch: An electromechanical switch triggered by motion or force, useful for tasks like counting, speed detection, and object sensing [4]. It has three terminals: Common, NO (Normally Open), and NC (Normally Closed), operating like a sensor to detect object presence/absence.

GSM Module: These connect devices to GSM-GPRS networks, enabling communication. GSM (Global System for Mobile Communications) is widely used, while GPRS (Global Packet Radio Service) boosts data speeds [10]. A GSM/GPRS module includes a modem, power supply, and interfaces (e.g., RS-232, USB) [6].

GSM Modem: Functions like a mobile phone with a SIM card, allowing computers to use cellular networks. Common uses include SMS/MMS messaging and mobile internet access.

http://dx.doi.org/10.21622/ACE.2026.06.1.1955

It can be a standalone device or integrated into phones, with connectivity via serial, USB, or Bluetooth [5].

SIM7600CE-T 4G Shield: An Arduino-compatible module supporting 4G/3G/2G and GNSS positioning. It offers LTE CAT4 speeds (150 Mbps downlink, 50 Mbps uplink) and low-power SMS/data transmission, ideal for IoT applications [6].

Arduino Microcontroller (Atmega): They receive the input from the sensors, perform arithmetical operations on it, make decisions, trigger actuators, and display outputs on the LCD (Liquid Crystal Display). The Arduino Uno board uses an ATmega328p chip. The microcontroller comprises 14 digital pins and 6 analog pins, a 16MHz crystal oscillator, and connects to a computer via a USB connector. The digital pins can be used for both input and output, but they have only two states, HIGH or LOW. The Arduino IDE, an open-source platform where instructions are compiled and uploaded onto the Arduino Microcontroller through the Universal Serial Bus interface, is used to run the hardware. In-System Programming was employed on the Arduino board to code the ATmega328p microcontroller on the sensor unit board and the ATmega32 microcontroller on the main controller (ISP). Also, in using an Arduino, it is easy to connect with other micro-controllers [7].

I2C LCD Display Module: The character display on the HD44780 platform and the I2C LCD adapter make up this robust, efficient display module. A total of 32 ASCII characters, grouped into two rows of 16 characters each, can be seen on its LED-backlit display. I2C is the interface employed for communication by this 16x2 LCD screen. It signifies that the display absolutely uses the four pins VCC, GND, SDA, and SCL. It will spare at least 4 Arduino digital or analog pins. Also, the Gravity interface makes it simpler to utilize for carrying out the project design.

Wi-Fi Module (ESP8266): An open-source Internet of Things framework called Esspressif runs on the ESP8266 Wi-Fi SoC. (System-on-Chip for Wi-Fi). Using the ESP8266, it is conceivable to host a program as well as assign every Wi-Fi networking function to another application processor. The ESP8266 module is pre-programmed with an AT command set firmware, allowing it to connect to the Arduino Microcontroller with only a few clicks and get about the same level of Wi-Fi functionality as a Wi-Fi Shield [8]. Its primary function is to link these systems to the Internet. This Wi-Fi has the following features: Compactness, High durability, Low power consumption, 100m Amps, +19.5dBm output power in 802.11b mode, and integrates an Integrated TCP/IP protocol stack. The ESP8266 needs a 3.3V to function accurately; any higher voltage will burn it or damage it. A voltage divider was introduced into the system for the required voltage by using the values of a selected resistor and engaging in the calculation of the Voltage divider:

$$Vout = Vin \,.\, \frac{R2}{R1+R2}$$

$$Vout = 5v \,.\, \frac{2kohms}{1kohms+2kohms}$$

$$Vout = 5v \,.\, \frac{2kohms}{3kohms}$$

$$Vout = 5v\ (0.6667)$$

$$\text{(approx.)}\ Vout = 3.3V$$

Energy Current Sensor: As a result, cutting cables or reconnecting circuits is no longer necessary. To read the current AC value, just clamp the AC transformer probe to the AC

http://dx.doi.org/10.21622/ACE.2026.06.1.1955

line and connect the 3.5mm headphone connector to the signal conversion module. A 3V3 or 5V microcontroller is intended to be compatible with the analog output. It is practical for measuring and monitoring AC current. The following characteristics apply to this research project: high safety, several ranges for diverse monitoring settings, and interoperability with 3V3 or 5V microcontrollers like the Arduino microcontroller. Non-contact monitoring.

Relay Module: Relays are switches that permit and prohibit devices in response to Arduino commands. There are three contractors in a fundamental relay: a normally open (NO), a normally closed (NC), and a common (COM). The COM is coupled with NC in the ON input state. A relay module's operational voltage ranges from 0V to 5.0V DC.

Power Source: The ATmega328 microcontroller operates at a voltage of 0V to 5.0VDC, and a voltage higher than 5.5V would burn the chip out. To establish this system, a rechargeable 12V DC battery is connected to multiple-output DC–DC converters, which are essential in this research for a multitude of applications where different DC output voltages are required. This voltage regulator is used to establish the connection. A 12V rechargeable battery is used to actualize this circuit. A 12V charger is used to charge the 12V battery [10].

Energy Meter Reading

For every kilowatt per hour, the watt-hour LED within the smart energy meter glows at different periods. Using an LED indicator, we can assess how worn out an office is. The electrical inspiration that develops when force is expended is used by these LED squints. The optocoupler PC817 IC is bundled with this drive. Due to the microcontroller's automated PIN commitment, the IC's yield is given. The number of major efforts can be established with the use of a counter, and the percentage of energy expended can then be calculated. The meter's beat is used to connect it to the microcontroller. Additionally, the fee per unit is calculated using the latency between beats. The essentiality meter illuminates the LED only once after each cycle. As a result, if a 100-watt bulb is used to light the fire, the flame will flash several times in a minute.

Mobile Alert

The microcontroller and GSM module are interconnected, and the GSM module serves as a channel for SMS messages with usage estimates for each day, and also to know if any tampering is occurring. With the flexible number that has been registered on the smart meter, the message may be sent daily near the end.

SOFTWARE DESIGN

According to where the microcontroller will be utilized, a suitable programming language ought to be adopted for the software's design and implementation [2]. Software development allows for the implementation of various programming languages. The analysis of the software requirements, which encompass the ability to interoperate with the system control unit and peripherals, the potential to translate an analog signal to machine code, the potential for managing the system, and the capabilities to choose a preference, ultimately determines the selection of programming language based on the design methodology[7].

Assembly language integration alongside C programming is the most effective for this design, grounded on the aforementioned specifications for the overall software design. Hence, we will discuss how to program the Arduino board, emphasizing the role of software in system operation. With CodeBlocks, the Arduino UNO is created. An original open-source Arduino program is termed Arduino IDE. A USB cable is used to link the microcontroller to the computer. The Arduino's ADC tracks shifts in the analog input voltage and translates them into a binary value between 0 and 1023 bits. It returned a number between 0 and 1023 via an analog-to-read output and evaluated it using the output voltage.

http://dx.doi.org/10.21622/ACE.2026.06.1.1955

CIRCUIT DIAGRAM

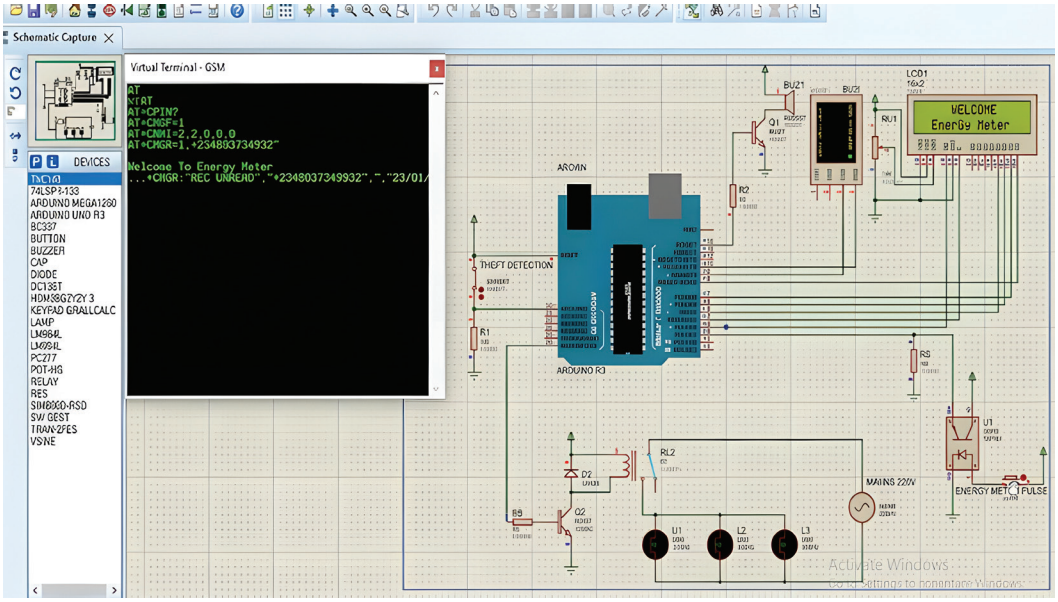


Figure 2: Circuit Diagram

The circuit design shown in Figure 2 is made using the hardware schematic shown in Figure 1. The software employed to build the circuit schematic and run the circuit simulation is Proteus Design Suite version 8.8. Printed circuit board schematics and designs are developed using the Proteus software tool, which is used for the electrical and electronic design (PCB). In the creation of electrical circuit testing, it is used. Automating the process of integrating components helps to conserve time and resources that might otherwise be lost. For example, unlike a physical component that could be damaged, if a component is connected incorrectly, it will only produce errors. The hardware system's various components were chosen from the Proteus development package's component menu. Each component is chosen and dragged to its proper location in the development environment. The GSM module, Relay module, and I2C LCD display are linked to the Arduino microcontroller's primary component through the digital pins depicted in Figure 2. The Arduino analyses the signal obtained from the energy meter via the tamper switch and transmits it to the GSM module and the ESP8266 Wi-Fi transmitter. From there, the data is transmitted to the cloud for the user to view and monitor, and then to the mobile devices of the designated users to monitor and also prevent energy theft. Finally, the receiver is to receive information from the transmitter and then process the information that it receives on a serial monitor. To provide an interface for reading values during simulation, the serial monitor is linked to pins 0 and 1. The components that will be used in the design, which have been correctly evaluated and researched, are all illustrated by the circuit diagram above.

FLOW-CHART DETAIL

This research work makes use of the prime cloud-based platform, which is an open-source program. A GSM module and an ESP8266 module with the tamper sensor can be used to send the sense of the theft through the tamper switch on the meter and send it to the user's designated device for assessment and control. Updates of all information are discovered throughout the day through the user's device to check the energy metering status. The user-designated numbers are first included in the system program to track the system's multiple indicators before making use of these features. The system's operational flow chart is shown in Figure 3.

http://dx.doi.org/10.21622/ACE.2026.06.1.1955

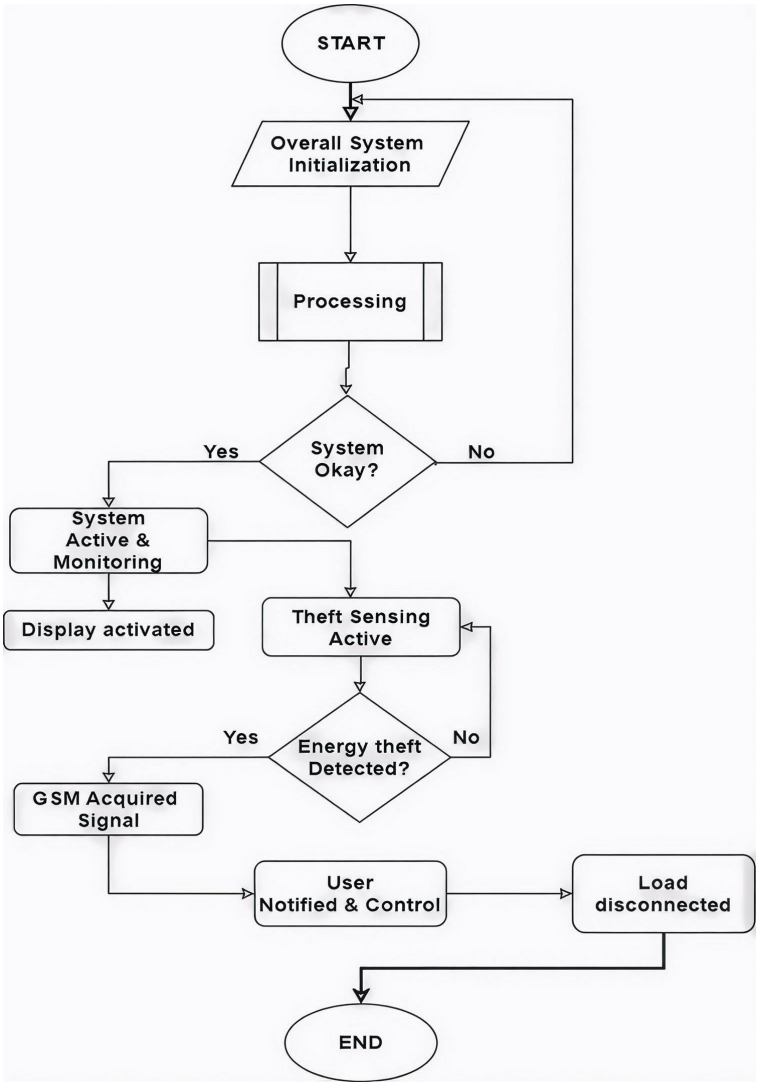


Figure 3: Flow-Chart Diagram

With the system design in place, the next step was to implement the system and evaluate its performance. This section details the implementation process and presents the results obtained.

4. SYSTEM IMPLEMENTATION AND RESULT

a) Hardware Implementation

This involves the building and synchronization of the physical components, sensors, power supplies, and controllers that make up the system's hardware. The breadboarding of the components, the mounting and connecting of components, and the packaging are all included in the physical execution of the project:

1.) Breadboarding of components

The different parts were breadboarded before the project was built, as shown in Figure 4, to ensure their functionality and that the circuit diagram is working accurately. In this stage, different components were tested, and the best ones suited to the use of the project were used. The testing of the hardware part of the project was done with the use of a breadboard before the components were further soldered to a continuous type Vero board, as revealed in Figure 4.

http://dx.doi.org/10.21622/ACE.2026.06.1.1955

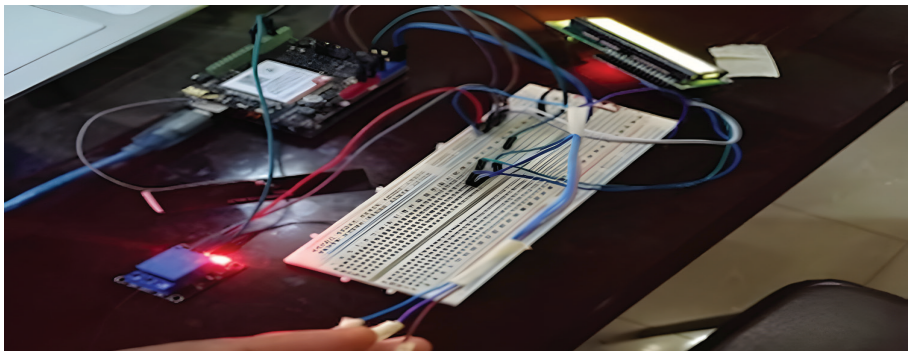


Figure 4: Components' Functionality testing on the Breadboard platform

2.) Implementation and Integrations of Components

Each component was put on the board and cemented on the surface, as shown in Figure 5, after they were tested to operate well with each other and produce desired outcomes with a high level of accuracy. After that, tests were performed to check that all of the components were in working order and had not been damaged during the installation. It should be noted that the connection and bonding operation was carried out in stages to address errors early on.

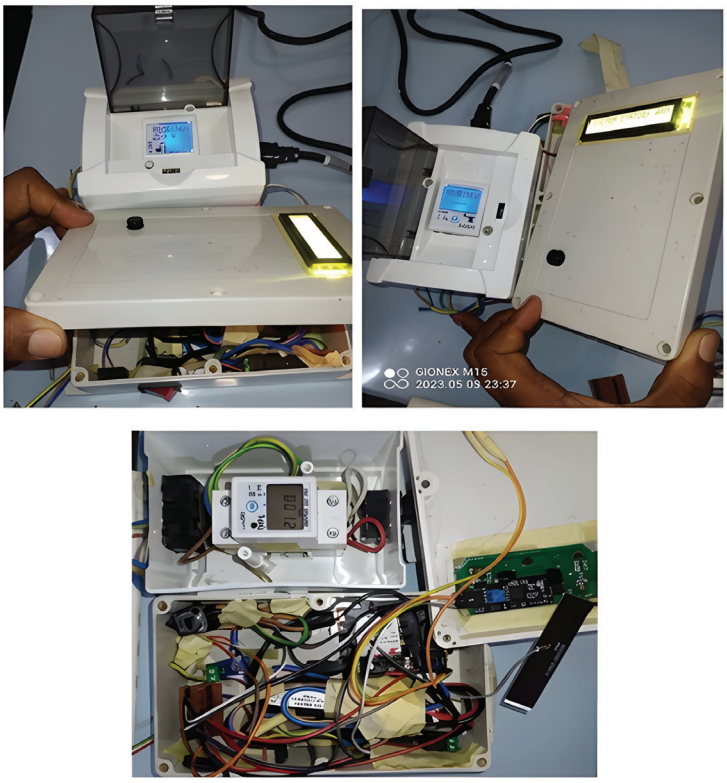


Figure 5: installation of components in a project package for meter monitoring and detection

3.) Packaging

The project was packaged in a particle board, as shown in Figure 4, after the mounting and soldering of various components. An arrangement was made to give the project a compact size. In an attempt to provide an internal power source for the controller, the GSM communication module, and other project-related sensors, a 12-volt DC lithium-ion battery was attached to the package box. Also, an internal pathway was made for a connector to

http://dx.doi.org/10.21622/ACE.2026.06.1.1955

be linked to a DC power source. The use of the intelligent power system kept the total build and construction fairly compact.

b) Software Implementation

It is comprised of the system's functionality, which also comprises the Proteus 8 Professional, the Arduino compiled code utilized to code it, and the embedding of the project into an application.

For this research work, the Arduino IDE was employed. The system's pre-installed Arduino IDE was upgraded to the most recent release, Arduino 1.8.15. The essential libraries, as well as the Adafruit GSM module and ESP8266 libraries, were successfully updated. The initial execution of the code was performed using the setup as well as the program development workspace, and subsequent runs were performed using the loop. It was run and compiled, and the bugs were fixed in the Arduino microcontroller. Sometimes, when applicable, the # include function was used to use the libraries.

During this research, the hardware block diagram layout shown in Figure 1 was employed in order to establish the appropriate location of components and their connections, as well as the viability of the previously established design. After designing an IoT-based energy meter monitoring and theft sensing system circuit, simulations were performed in Proteus to enhance the circuit parameters that preceded the hands-on construction of the system design. The complete diagram of the circuit is revealed in Figure 2, displaying all critical components before the circuit's wiring. The Proteus schematic capture and component mode from the library's toolbar were used to select each component, although some libraries were nonexistent.

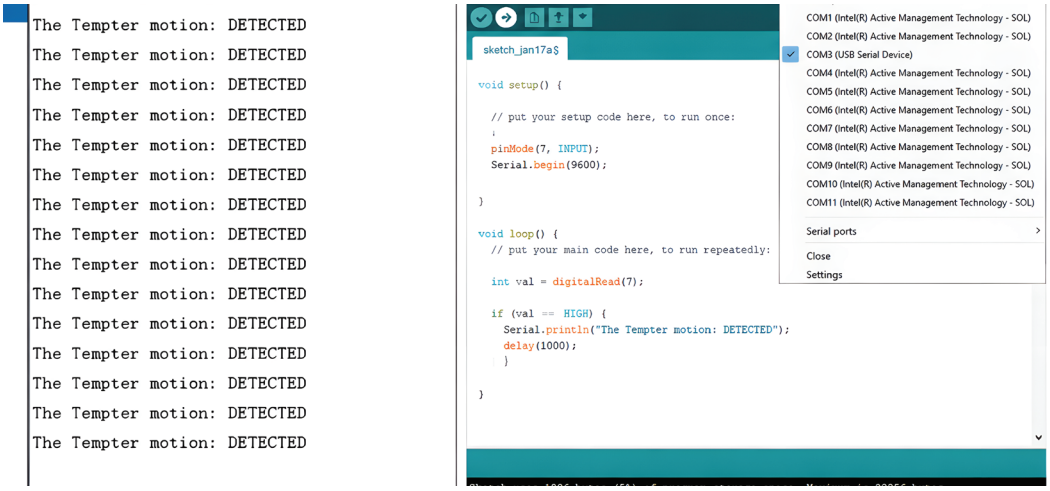


Figure 6: Software performance testing

c. System and Result

Electronics testing is a crucial stage in the design and implementation of systems. Unit testing, program evaluation, subsystem evaluation, system testing, and validation are just a few examples of the numerous test kinds performed in this work. Before performing a comprehensive overall system test, the sensor parts, subsystems, and hardware were evaluated.

i) Unit testing: The IoT Smart Energy Meter Monitoring and Energy Theft Sensing

System components were first tested for continuity and power rating with the aid of a multimeter to ensure their functionality before integration into the system. Tests were carried out on the power supply and the microcontroller.

ii) Subsystem testing: The subsystem test was carried out to ensure each unit was

http://dx.doi.org/10.21622/ACE.2026.06.1.1955

functional while gradually integrating the units. This test was conducted while gradually developing the system; the sensors were tested using Arduino and other components to ensure optimal functionality. It also involves the correct placement of the components to determine their voltage and current consumption.

iii) System testing: During system testing, the IoT Smart Energy Meter Monitoring and

The Energy Theft Sensing System is tested as an entire unit and as a unified system. It was at this stage that it was discovered the transceivers were not supplied with the required voltage.

Figure 7: Final Package of the Meter and the Sensing Display box.

iv) Acceptance testing: The IoT Smart Energy Meter Monitoring and Energy Theft Sensing

Systems were connected to a voltage source and operated by an individual after the construction had been completed.

v) Program Testing: After developing, compiling, and debugging the program code to

ensure there are no errors, the code is then uploaded onto the microcontroller and tested. Some libraries were missing and had to be downloaded and added before the code could function appropriately.

vi) System's Accuracy in Detecting Energy Theft: In a series of tests conducted over a

For a period of 30 days, the IoT Smart Energy Meter Monitoring and Energy Theft Sensing System was evaluated for its accuracy in detecting energy theft. A total of 100 simulated theft scenarios were introduced, with different methods of unauthorized energy consumption, such as bypassing the meter and tampering with the wiring. The system successfully detected 95 out of the 100 theft scenarios, resulting in an accuracy rate of 95%. The false - positive rate was 3%, with 3 instances where the system incorrectly flagged normal energy usage as theft. These results are summarized in Table 2:

Table 2: System's Accuracy in Detecting Energy Theft

S/N	Test parameter	VALUES
1	Total number of theft scenarios	100
2	Number of correctly detected thefts	95

http://dx.doi.org/10.21622/ACE.2026.06.1.1955

3	Accuracy rate	95%
4	False - positive rate	3%

The response time for remote control actions was measured using a mobile device to send commands to the smart meter. A total of 50 remote control commands, including turning off the meter and adjusting energy consumption settings, were sent. The average response time was found to be 5 seconds, with a minimum response time of 3 seconds and a maximum response time of 8 seconds. The response time distribution is shown in Figure 8:

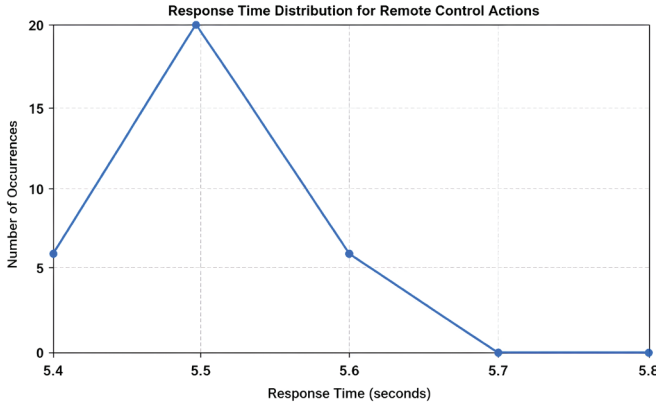


Figure 8: Quantitative data on system performance

vii) Power consumption: The power consumption of the system was measured over a 24-hour period

hour period. The components of the system, including the Atmega microcontroller, GSM module, and sensors, were monitored. The total power consumption was 12 Wh (watt - hours) per day. The breakdown of power consumption by component is as follows:

Table 3: Power consumption details

COMPONENT	POWER CONSUMPTION PER DAY (Wh)
Atmega microcontroller	4
GSM module	5
Sensors (tamper sensor, AC current sensor, etc.)	3

5. NOVEL CONTRIBUTION AND DISTINCTION FROM EXISTING SYSTEMS

The contribution of this paper to IoT-based smart energy metering is by integrating real-time theft detection, dual communication architecture, and automated control mechanisms into a unified system. While previous studies have implemented IoT-based smart meters using GSM or Wi-Fi independently, the proposed system combines both communication channels to enhance reliability and redundancy.

The novelty of this research lies in the following:

1. Hybrid Communication Framework
2. Unlike conventional systems that rely solely on GSM or Wi-Fi, this system integrates both ESP8266 (Wi-Fi) and GSM modules that ensure continuous data transmission even in cases of network failure.

http://dx.doi.org/10.21622/ACE.2026.06.1.1955

3. Integrated Theft Detection with Automatic Disconnection
4. Existing systems often focus on monitoring and alerting. This system incorporates real-time tamper detection coupled with automatic relay-based power disconnection and actively prevents energy theft.
5. Remote User Control via GSM Interface
6. The system allows users to not only monitor consumption but also remotely control energy supply, which extends beyond basic monitoring functions reported in earlier works.
7. Low-Cost and Scalable Design
8. The architecture is designed using widely available components such as ATmega328p and ESP8266, which makes it suitable for deployment in developing regions with cost constraints.

6. CONCLUSION

This research developed an IoT - based smart meter monitoring system to tackle energy management and security issues in developing countries. The system, incorporating an Atmega microcontroller, GSM module, and non - invasive sensor, offers key features like theft detection and remote control. Upon detecting theft, it alerts users and cuts power, with high accuracy in testing. Remote monitoring enables real - time energy usage access, empowering users to manage consumption. Rigorous testing during implementation verified its functionality. However, scalability and power consumption optimization remain areas for future research. Overall, the system provides an effective energy management solution, with potential for broader impact on sustainable energy use as it evolves.

7. DECLARATIONS

AUTHOR CONTRIBUTION

The authors carried out the whole research.

FUNDING

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

COMPETING INTEREST

There is no competing interest.

CONSENT TO PUBLISH DECLARATION:

The authors consent to the publication of this paper. The authors have read and approved the final version of the manuscript and agreed to its submission. No part of this paper has been published or is under consideration elsewhere.

REFERENCES

[1] M. H. Yaghmaee and H. Hejazi, "Design and Implementation of an Internet of Things Based Smart Energy Metering," in *2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, IEEE, Aug. 2018, pp. 191–194. doi: 10.1109/SEGE.2018.8499458.

[2] E. Adetiba *et al.*, "Development of an IoT Based Data Acquisition and Automatic Irrigation System for Precision Agriculture," in *2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON)*, IEEE, Apr. 2022, pp. 1–5. doi: 10.1109/NIGERCON54645.2022.9803132.

[3] M. Tahir, N. Ismat, H. H. Rizvi, A. Zaffar, S. M. Nabeel Mustafa, and A. A. Khan, "Implementation of a smart energy meter using blockchain and Internet of Things: A step toward energy conservation," *Front. Energy Res.*, vol. 10, Dec. 2022, doi: 10.3389/fenrg.2022.1029113.

http://dx.doi.org/10.21622/ACE.2026.06.1.1955

[4] S. G. Rajeshwari, N. K. Kalkeri, N. Pati, S. Jalihal, and S. Barigali, "IOT based smart energy meter for efficient energy utilization and billing," I*nt. J. Adv. Res. Ideas Innov. Educ. (IJARIIE)*, vol. 8, no. 4.

[5] A. Rachedi, M. H. Rehmani, S. Cherkaoui, and J. J. P. C. Rodrigues, "IEEE Access Special Section Editorial: The Plethora of Research in Internet of Things (IoT)," *IEEE Access*, vol. 4, pp. 9575–9579, 2016, doi: 10.1109/ACCESS.2016.2647499.

[6] "SIM7600G-H CAT4 4G Shield for Arduino | Ultimate Connectivity," 2025. [Online]. Available: https://wiki.dfrobot.com/SIM7600CE-T_4G(LTE)_Shield_V1.0_SKU_TEL0124

[7] A. E. Akindele, A. I. Afolabi, V. W. Oguntosin, O. A. Alashiri, V. O. Matthews, and O. O. Akande, "Development of an intelligent smart shopping Cart system," in *Lecture Notes in Engineering and Computer Science*, 2019.

[8] B. K. Barman, S. N. Yadav, S. Kumar, and S. Gope, "IOT Based Smart Energy Meter for Efficient Energy Utilization in Smart Grid," in *2018 2nd International Conference on Power, Energy and Environment: Towards Smart Technology (ICEPE)*, IEEE, Jun. 2018, pp. 1–5. doi: 10.1109/EPETSG.2018.8658501.

[9] N. Agnihotri, "GSM/GPRS Module : All You Need To Know," 2022. [Online]. Available: https://www.engineersgarage.com/gsm-gprs-module-all-you-need-to-know/

[10] A. R. and K. V., "IoT Based Smart Energy Theft Detection and Monitoring System for Smart Home," in *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, IEEE, Jul. 2020, pp. 1–6. doi: 10.1109/ICSCAN49426.2020.9262411.

http://dx.doi.org/10.21622/ACE.2026.06.1.1933

LOW-RESOURCE FINE-TUNING OF LLAMA-3.1 USING QLORA FOR NIGERIAN LINGUISTIC CONTEXTS

Solomon Joseph Udoabba[1], Bliss Utibe-Abasi Stephen[1,2*], Oluseun Damilola Oyeleke[3], Philip Asuquo[1,2] and Sadiq Thomas[4]

[1]University of Uyo, Department of Computer Engineering, Nigeria
[2]TETFund Centre of Excellence in Computational Intelligence Research
[3]Global Banking School, United Kingdom
[4]Nile University, Department of Computer Engineering, Abuja, Nigeria

Emails: { blissstephen@uniuyo.edu.ng, uniqueudoabba@gmail.com, contactseun@gmail.com, philipasuquo@uniuyo.edu.ng, sadiqthomas@nileuniversity.edu.ng }

Received 31 March 2026 - Accepted 20 May 2026 - Published 23 June 2026

ABSTRACT

Large language models (LLMs) struggle to accommodate underrepresented linguistic contexts, such as Nigerian English and Nigerian Pidgin, due to Western-centric training corpora. While fine-tuning offers a solution, aggressive adaptation strategies may risk degrading general-domain performance when trained on small datasets.

This study investigates the feasibility of a conservative Quantized Low-Rank Adaptation (QLoRA) strategy to align LLaMA-3.1-8B with Nigerian contexts under extreme compute constraints. As a pilot study, we employed a dampened update scaling factor (α = 16, r = 64) on a stratified subset of 12,000 Nigerian web text samples. This configuration was deliberately chosen to prioritize model stability and preserve pre-trained general knowledge while inducing cultural alignment. Preliminary results indicate that this conservative approach successfully avoids model collapse, yielding a stable reduction in perplexity (8.20 to 7.38). Initial qualitative probing reveals a divergence between pragmatic alignment and semantic precision: the model successfully internalized the affective sentiment of Nigerian slang (e.g., associating "Sapa" with frustration), even where it lacked sufficient data to generate precise dictionary definitions. Furthermore, we discuss the economic implications of this approach, arguing that parameter-efficient fine-tuning offers a lower "inference tax" forresource-constrained deployment compared to few-shot prompting. This work provides a technical reference for "safety-first" adaptation in low-resource academic environments.

We explore whether large language models can be meaningfully adapted to Nigerian linguistic contexts using limited computational resources. Many African languages and regional dialects are underrepresented in the data used to train modern foundation models. As a result, these systems may misinterpret culturally grounded expressions or fail to capture local usage patterns. In this work, we fine-tuned LLaMA-3.1-8B using QLoRA on Nigerian web text while operating under hardware constraints typical of universities in developing regions. Instead of pursuing large-scale optimization, we focused on feasibility, training behavior, and qualitative response shifts.

Our results show that parameter-efficient fine-tuning can introduce measurable contextual adaptation without requiring full model retraining. At the same time, improvements are modest and do not eliminate semantic inaccuracies. By presenting both strengths and limitations, we aim to provide a realistic foundation for future work on culturally aligned AI systems in low-resource settings.

Keywords: *Parameter-efficient fine-tuning, natural language processing, low resource languages, Nigerian Pidgin*

http://dx.doi.org/10.21622/ACE.2026.06.1.1933

INTRODUCTION

Artificial Intelligence has emerged as a dominant computational paradigm in Natural Language Processing (NLP), with models such as OpenAI's GPT series and Meta's LLaMA series setting new performance benchmarks [1]. Specifically, Meta LLaMA 3.1 8B offers a powerful foundation for downstream applications due to its balance of performance and efficiency [2]. However, the efficacy of these models is inextricably linked to their training data, which remains predominantly Western-centric.

This data disparity creates a "representation gap" for African languages. In Nigeria, a nation with over 520 languages and a unique lingua franca (Nigerian Pidgin), standard models often fail to capture syntactic and cultural nuances [3]. For instance, cultural slang terms like "Sapa" (financial struggle) or "Jollof wars" are frequently misinterpreted by base models as generic English terms. This limitation hinders the development of effective NLP applications for local needs, from automated customer service to educational tools [4].

To address this, we investigate the fine-tuning of LLaMA 3.1 8B for Nigerian contexts using Parameter-Efficient Fine-Tuning (PEFT). Recent works by Hu et al. [5] and Dettmers et al. [6] have demonstrated that QLoRA (Quantized Low-Rank Adaptation) allows for the fine-tuning of massive models on single GPUs without significant performance degradation. Similar methodologies have been successfully applied to Vietnamese healthcare [7] and South African languages (Zulu/Xhosa) [8], validating PEFT as a viable pathway for low-resource adaptation.

The central contribution of this work is a reproducible, low-cost methodology for adapting LLMs to Nigerian linguistic contexts. We hypothesize that by employing QLoRA on a curated subset of the NaijaWeb corpus, we can induce measurable shifts in the model's cultural alignment and Pidgin fluency within a resource-constrained computational environment.

It is important to note that this work is positioned as an exploratory feasibility study rather than a comprehensive benchmark evaluation. Due to computational and dataset constraints, the objective is to examine whether parameter-efficient fine-tuning can produce measurable shifts in cultural alignment within a limited-resource academic setting. Accordingly, the evaluation focuses on preliminary quantitative indicators and structured qualitative assessment rather than large-scale comparative benchmarking.

MATERIALS AND METHODS

DATASET PREPARATION

The training data was sourced from the NaijaWeb corpus [9], a diverse collection of web-scraped Nigerian text. To accommodate the memory constraints of our hardware (Tesla T4 16GB VRAM), we selected a stratified subset of 12,000 samples. The data distribution included Nigerian English (45%), Nigerian Pidgin (30%), and code-switched text (25%).

Crucially, we reformatted the raw text into an instruction-following format (Alpaca style) to align with the supervised fine-tuning (SFT) paradigm. Each entry was structured as an input instruction (e.g., "Explain this concept") and a target output,

http://dx.doi.org/10.21622/ACE.2026.06.1.1933

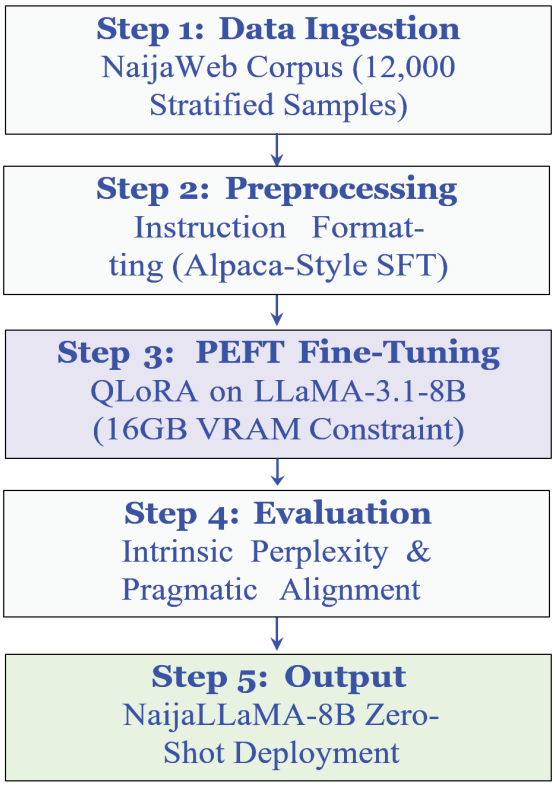


Figure 1: Proposed Methodology Framework.

The workflow illustrates the end-to-end pipeline utilized in this study: (1) Ingestion of the NaijaWeb Dataset, (2) Stratification and Alpaca-style instruction preprocessing, (3) Parameter-efficient fine-tuning via QLoRA under a 16GB VRAM constraint, (4) Model evaluation focusing on perplexity and qualitative pragmatic alignment, and (5) The proposed zero-shot deployment model. facilitating the model's ability to follow cultural prompts. A separate validation set of 500 examples was reserved for evaluation.

MODEL CONFIGURATION AND HARDWARE

We utilized unsloth/llama-3.1-8b-bnb-4bit, a pre-quantized version of Meta's LLaMA 3.1. The use of 4-bit NormalFloat (NF4) quantization was essential to fit the 8-billion parameter model into the 16GB VRAM of a single GPU provided by the Kaggle platform.

We employed the Unsloth library for optimization, which provides up to 2x faster training speeds and reduced memory fragmentation compared to standard Hugging Face implementations.

FINE-TUNING HYPERPARAMETERS

We applied QLoRA adapters to the linear layers of the attention mechanism (q proj, k proj, v proj, o proj). The specific hyperparameters were:

- LoRA Rank (r): 64
- LoRA Alpha (α): 16 (Conservative scaling to prevent overfitting)
- Dropout: 0

http://dx.doi.org/10.21622/ACE.2026.06.1.1933

- Optimizer: AdamW 8-bit
- Learning Rate: 2×10^{-4} with linear decay
- Batch Size: 2 (with 8 gradient accumulation steps for an effective batch size of 16)

The model was trained for 1 epoch to prevent overfitting on the small dataset, with checkpoints saved every 50 steps.

RESULTS

As shown in Fig 2, the training loss decreased steadily from 2.04 to 1.98, indicating stable optimization under conservative scaling. The model's adaptation was evaluated using both quantitative metrics and qualitative human evaluation to measure cultural alignment.

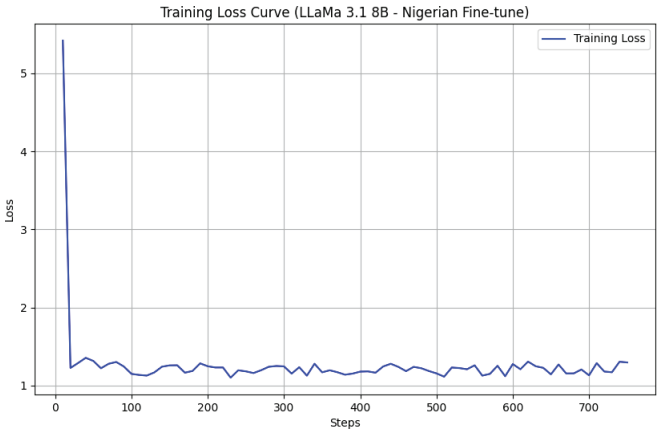


Figure 2: Training loss trajectory during QLoRA fine-tuning.

The model was trained for one epoch on 12,000 Nigerian web samples using conservative scaling ($r = 64$, $\alpha = 16$). The smooth decline indicates stable optimization dynamics during fine-tuning. While validation loss was not extensively tracked due to compute constraints, no instability was observed during training.

QUANTITATIVE EVALUATION

We observed improvements across intrinsic metrics, as detailed in Table 1. The reduction in perplexity indicates that the model became less "surprised" by Nigerian sentence structures and Pidgin syntax.

Table 1: Intrinsic Evaluation Metrics.

Metric	Baseline (LLaMA 3.1)	Fine-Tuned (NaijaLLaMA)	Improvement
Training Loss	2.04	1.98	-2.9%
Perplexity	8.20	7.38	-10.0%

Lower values indicate better performance.

STRUCTURED QUALITATIVE EVALUATION

To reduce anecdotal bias, a lightweight structured evaluation was conducted on 30 prompts spanning three categories: (1) Nigerian slang interpretation, (2) Nigerian Pidgin translation, and (3) culturally grounded knowledge queries. Prompts were sampled from culturally relevant expressions present in the training corpus.

http://dx.doi.org/10.21622/ACE.2026.06.1.1933

Each response was manually assessed by two Nigerian postgraduate students familiar with Nigerian English and Pidgin. Outputs were categorized as Correct, Partially Correct, or Incorrect based on semantic accuracy and contextual appropriateness.

Across the 30 prompts, the fine-tuned model produced 18 Correct, 7 Partially Correct, and 5 Incorrect responses. In contrast, the base model produced 11 Correct, 9 Partially Correct, and 10 Incorrect responses. While not statistically analyzed, this distribution suggests a consistent directional shift toward improved contextual alignment in culturally specific tasks. Formal statistical validation with larger evaluation sets remains necessary to confirm robustness.

These findings should be interpreted as preliminary due to limited sample size and absence of formal inter-rater reliability measurement.

Table 2: Comparative Analysis of Model Generations.

Task	Prompt	Base Model Output	Fine-Tuned Output
Slang	Explain "Sapa"	"Sapa is a slang term... for something unimportant." (Hallucination)	"Sapa is a term... used to express disapproval or disappointment... frustration with a situation."
Translation	Translate "How are you?"	"How are you doing?" (Standard English)	"How you dey?" (Syntactically Correct Pidgin)
Knowledge	Who is Fela Kuti?	(Generic Biography)	"Fela Kuti was a Nigerian multi-instrumentalist... nickname was 'the Black President'."

The examples illustrate qualitative shifts in contextual grounding following adaptation.

DISCUSSION

The central objective of this study was to determine if a state-of-the-art foundation model could be meaningfully adapted to Nigerian linguistic nuances using

consumer-grade hardware and a "conservative" adaptation strategy. The results offer three critical insights into the dynamics of low-resource fine-tuning.

STABILITY VS. PLASTICITY IN HYPERPARAMETERS

A key methodological decision in this work was the selection of a LoRA effective scaling factor of 0.25 (computed as $\alpha/r = 16/64$). Many implementations use higher effective scaling factors to maximize adaptation strength. However, given the noisy nature of web-scraped text and the limited dataset size (12,000 samples), we hypothesized that aggressive updates would risk overfitting or "catastrophic forgetting" of the model's core logic.

Our stable training loss trajectory confirms that the dampened signal acted as an effective regularizer. The model did not exhibit the erratic behavior often associated with high-rank updates on small data. This suggests that for low-resource languages where high-quality, instruction-verified data is scarce, a low-alpha strategy may provide a safer pathway to adaptation, effectively "nudging" the model's tone without overwriting its foundational reasoning capabilities.

PRAGMATIC ALIGNMENT VS. SEMANTIC DEFINITION

The model's performance on cultural concepts, particularly the term "Sapa," illuminates a distinction between pragmatic alignment and semantic precision.

http://dx.doi.org/10.21622/ACE.2026.06.1.1933

When asked to define "Sapa" (a slang term for financial struggle), the model described it as a term for "disapproval or frustration." While semantically inaccurate (it missed the specific definition of "poverty"), the model successfully captured the sentiment polarity. It understood that "Sapa" appears in distributions of text

associated with negative sentiment and complaint, even if it could not retrieve the precise dictionary definition. This phenomenon suggests that QLoRA on small, unannotated corpora is highly effective at Association Learning (capturing the "vibe" or emotional context) but less effective at acquiring new factual definitions without explicit instruction tuning. Researchers using scraped data should therefore expect improvements in tone before improvements in factual accuracy.

INFERENCE EFFICIENCY AND DEPLOYMENT VIABILITY

While recent literature suggests that Few-Shot Prompting (In-Context Learning) can achieve comparable accuracy to fine-tuning, we argue that fine-tuning may offer practical deployment advantages in settings with limited bandwidth or compute due to Inference Economics.

Few-shot prompting requires feeding long context examples with every single query, significantly increasing the token count, latency, and financial cost per interaction. In contrast, our PEFT approach "bakes" the cultural context into the adapter weights.

This allows for zero-shot deployment, where the model understands Nigerian context without requiring expensive system prompts. For deployment in rural or academic settings with limited internet bandwidth and hardware, this reduction in inference-time compute is a critical advantage that outweighs marginal gains in accuracy.

LIMITATIONS

We acknowledge that the conservative scaling factor ($\alpha = 16$) likely limited the total possible reduction in perplexity. Future work with larger, curated datasets should explore dynamic α scheduling to balance stability with more aggressive learning.

Additionally, while the model captures the syntax of Nigerian Pidgin, semantic hallucinations persist, indicating the need for the incorporation of curated lexical definition data in future dataset preparation.

CONCLUSION

In this work, we refer to the fine-tuned variant as NaijaLLaMA-8B for clarity, while acknowledging that it consists of LoRA adapters applied to the LLaMA-3.1-8B base model. By leveraging Unsloth and QLoRA, we adapted a state-of-the-art 8B parameter model on a single GPU, indicating preliminary improvements in culturally contextualized responses and Pidgin fluency.

Future work will focus on these areas: (1) scaling the dataset to include explicit definitions of slang to improve semantic precision; and (2) extending the training to include low-resource indigenous languages like Ibibio and Tiv.

http://dx.doi.org/10.21622/ACE.2026.06.1.1933

SUPPORTING INFORMATION

S1 File. Training Logs and Model Adapters. The full training logs, loss curves, and the trained LoRA adapters are available at https://huggingface.co/luminaudoabba/llama3-naijaweb-merged.

ACKNOWLEDGMENTS

We acknowledge the Department of Computer Engineering, University of Uyo, for providing the academic environment to conduct this research. We also thank the creators of the NaijaWeb corpus and the Unsloth AI team for their open-source contributions.

REFERENCES

[1] L. Qin *et al.*, "Large Language Models Meet NLP: A Survey," *arXiv.org*, 2024. https://arxiv.org/abs/2405.12819

[2] A. Dubey *et al.*, "The Llama 3 Herd of Models," *arXiv.org*, 2024. https://arxiv.org/abs/2407.21783

[3] I. Inuwa-Dutse, "NaijaNLP: A Survey of Nigerian Low-Resource Languages," *arXiv.org*, 2025. https://arxiv.org/abs/2502.19784

[4] J. Ojo *et al.*, "AfroBench: How Good are Large Language Models on African Languages?," *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 19048–19095, Jan. 2025, doi: https://doi.org/10.18653/v1/2025.findings-acl.976.

[5] E. J. Hu *et al.*, "LoRA: Low-Rank Adaptation of Large Language Models," *arXiv:2106.09685 [cs]*, Oct. 2021, Available: https://arxiv.org/abs/2106.09685

[6] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs," *arXiv.org*, May 23, 2023. https://arxiv.org/abs/2305.14314

[7] N. Bui *et al.*, "Fine-tuning Large Language Models for Improved Health Communication in Low-Resource Languages," *Computer Methods and Programs in Biomedicine*, vol. 263, pp. 108655–108655, Feb. 2025, doi: https://doi.org/10.1016/j.cmpb.2025.108655.

[8] P. W. Khoboko, V. Marivate, and J. Sefara, "Optimizing translation for low-resource languages: Efficient fine-tuning with custom prompt engineering in large language models," *Machine Learning with Applications*, vol. 20, p. 100649, Jun. 2025, doi: https://doi.org/10.1016/j.mlwa.2025.100649.

[9] S. A. Ayanniyi, "Naijaweb: A Web Scraped Nigerian Context Dataset," *Huggingface.co*, 2021. https://huggingface.co/datasets/saheedniyi/naijaweb

ACE

ADVANCES in COMPUTING & ENGINEERING

JOURNAL

VOLUME 6
- ISSUE 1 -
JUNE 2026

E-ISSN:2735-5985