Enhancing Cybersecurity with Random Forest: Efficient Detection of Cyberattacks

ARTICLE HISTORY
Received 26 February 2026
Accepted 3 June 2026
Published 7 July 2026

Phathutshedzo Cyprin Ramuhovhi
Vaal University of Technology
Computer Sciences and Engineering Vanderbijlpark, South Africa
214102653@edu.vut.ac.za
ORCID: 0000-0001-9232-8852

Naume Sonhera
Vaal University of Technology
Computer Sciences and Engineering
Vanderbijlpark, South Africa
nqume@vut.ac.za
ORCID: 0000-0002-8275-2016

Tranos Zuva
Vaal University of Technology
Computer Sciences and Engineering
Vanderbijlpark, South Africa
tranosz@vut.ac.za
ORCID: 0000-0001-9579-3899

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2026

P. Ramuhovhi, N. Sonhera, and T. Zuva
"Enhancing Cybersecurity with Random Forest: Efficient Detection of Cyberattacks",
Latin-American Journal of Computing (LAJC), vol. 13, no. 2, 2026.

Enhancing Cybersecurity with Random Forest: Efficient Detection of Cyberattacks

Phathutshedzo Cyprin Ramuhovhi
Vaal University of Technology
Computer Sciences and Engineering
Vanderbijlpark, South Africa
214102653@edu.vut.ac.za

Naume Sonhera
Vaal University of Technology
Computer Sciences and Engineering
Vanderbijlpark, South Africa
nqume@vut.ac.za

Tranos Zuva
Vaal University of Technology
Computer Sciences and Engineering
Vanderbijlpark, South Africa
tranosz@vut.ac.za

***Abstract*—** The rapid increase in the number of cyberattacks in the digital age has reduced the effectiveness of conventional cybersecurity systems. Traditional methods of cybersecurity face considerable difficulty when detecting new, sophisticated attacks and advanced exploitation techniques swiftly. This research addresses critical cybersecurity concerns by developing an AI-driven Intrusion Detection System (IDS), which employs Random Forest (RF) algorithms to detect cyberattacks efficiently. The evaluation of the model was conducted using three publicly available datasets: CICIDS2017 (692,703 records), NSL-KDD (148,517 records), and UNSW-NB15 (257,673 records) with various attack backgrounds and network configurations. A set of evaluation metrics, including accuracy, precision, recall, and F1-score, was employed to assess the performance of the cyberattack detection prototype. Across the three datasets, the model attained an average accuracy of 99.85%, precision of 99.83%, recall of 99.91%, and an F1-score of 99.87%, while maintaining low error rates, with an average false positive rate of 0.25% and a false negative rate of 0.10%. The results indicate that Random Forest is an effective solution for cyberattack detection in data-driven environments. The model was developed with lightweight and easy-to-deploy criteria, but the evaluation reported in this study was done under benchmark test conditions. This work improves the effectiveness of machine learning-based intrusion detection systems and serves as a stepping stone for future research on operational and real-time deployment of machine learning-based intrusion detection systems.

***Keywords*—** *Artificial Intelligence, Cybersecurity, Cyberattack, Intrusion Detection System*

I. INTRODUCTION

The rapid growth of the digital environment has elevated cybersecurity to a critical priority for both individuals and organizations. The frequency of cyberattacks has increased drastically over the last few decades to keep up with ever-changing technologies [1]. This increase requires organizations to implement strong cybersecurity measures. A study by [1] noted that traditional cybersecurity approaches, such as computer security and network protection systems, are becoming increasingly ineffective in combating continuously evolving and creative cyberattacks. Cyberattacks result in damaged reputations, financial losses due to the theft of intellectual property, legal liabilities, and business operation disruptions [2]. A cyberattack is a deliberate attempt by malicious actors to gain unauthorized access, disrupt operations, or compromise information systems, to destroy data, or perform any malicious activities that will compromise the company network or infrastructure [3]. In the study by [4], traditional cybersecurity is defined as methods such as conventional Intrusion Detection Systems (IDS) that operate based on signature-based and rule-based detection mechanisms in which cyberattacks are detected by matching the data against known attack patterns. The authors further emphasize that these methods are slowly becoming ineffective due to evolving, increasingly sophisticated cyberattacks, often supported or enhanced by Artificial Intelligence (AI) techniques and other advanced cyber capabilities. The world is experiencing an increase in cyberattacks, and the cost is estimated to go up to 10.5 trillion by 2025, compared to 3 trillion in 2015 [5]. In Africa, cases have increased by 76% in 2023, and South Africa has been the most affected by financial losses despite the high-level security measures [6]. Conventional cybersecurity methods, which rely on signature detection, are weak at dealing with advanced threats such as zero-day and polymorphic attacks [1]. Banks remain primary targets for cyberattacks [7], [8], [9] and the fast development of digital systems has increased vulnerabilities. According to recent research, AI-based models, such as SVMs and LSTMs, can increase detection through predicting and simulating attacks [2]. Nonetheless, most AI products are memory-consuming and cannot be used in real-time. The paper fills these gaps by suggesting an optimized, interpretable, and lightweight Random Forest (RF)-based IDS, which is meant to be used in real-time detection in resource-constrained settings, especially in areas such as South Africa, where cyberattacks have dire economic implications.

II. PROBLEM STATEMENT

Most businesses rely on digital technologies, which have resulted in a rise in advanced cyberattacks that conventional cybersecurity approaches fail to detect [2]. The same authors add that conventional cybersecurity approaches involve signature-based and rule-based detection models, which detect cyberattacks by matching data to known attack patterns. These methods often fail to detect new attacks. The complexity of cybersecurity attacks is constantly increasing, presenting serious difficulties for people and companies globally [10]. Traditional IDSs are critical security components that monitor and analyze system activity to detect suspicious behavior and prevent unauthorized access [4]. Research, such as that conducted on [6] and [10], points to AI models as more efficient than traditional algorithms. Deep Learning (DL) and machine learning (ML) models can achieve high accuracy level for real-time detection of zero-day attacks and advanced types of malware. The authors added

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2026
DOI: https://doi.org/10.33333/lajc.vol13n2.08

that DL models can achieve high accuracy, but they have high computational requirements. These studies show that traditional cybersecurity methods face significant challenges in detecting sophisticated attacks and advanced exploitation techniques in a timely manner. Thus, this paper delves into using AI to detect cyberattacks in real-time, aiming to improve accuracy and adaptability to emerging cyberattacks.

III. Related Works

Conventional IDSs are based on rule-based and signature-based techniques to identify known attack patterns, which are becoming less effective against contemporary and sophisticated threats [1], [4]. To address these shortcomings, AI technologies are being explored for real-time cyberattack detection, leveraging techniques such as anomaly detection, pattern recognition, and continuous learning[11], [12]. With the ability to provide adaptive and proactive defense, AI can provide the opportunity to identify and react to changing attacks in real-time, a shift from traditional static, signature-based security systems to dynamic, intelligent cybersecurity solutions.

A. The Emergence and Integration of Artificial Intelligence in Various Sectors

AI-based technologies, including machine learning, computer vision, and Convolutional Neural Networks (CNNs), are increasingly applied in agricultural settings to support real-time monitoring of plant conditions, detect diseases at the earliest stages, and manage water in irrigation, optimizing water usage and improving productivity [13]. Advanced deep learning has significantly improved the accuracy of medical image analysis, supporting the diagnosis of conditions such as pneumonia, tumors, and COVID-19, and AI-enabled virtual care systems, based on wearables and chatbots, enable continuous health monitoring and personalized treatment in healthcare [14]. Likewise, in education, AI promotes adaptive learning by having intelligent tutoring systems and automates grading, applying natural language processing (NLP) techniques to support automated grading. These technologies contribute to more personalized learning experiences while allowing educators to focus on innovative teaching practices [15].

AI has been shown to have a lot of transformative capability in diverse fields, such as education, agriculture, and healthcare, where it has increased the efficiency of running operations and decision-making [12], [13], [14]. In the information technology industry, in particular, AI solutions have become more advanced by enhancing complex data analysis rather than simple automation [16]. In this study, there is a special emphasis on the implementation of AI in the context of cybersecurity systems, where its development process is being leveraged to respond to new challenges in technology.

B. The Emergence and Integration of Artificial Intelligence in Cybersecurity

Increasing sophistication and pervasiveness of cyberattacks demand that organizational defense systems change their paradigm [17]. Traditional rule-based security measures are no longer effective when confronted with the adaptive and advanced nature of modern cyberattacks [1]. As a result, improving cybersecurity operations now requires the integration of AI. The main function of AI in enhancing cyber defense, as per the body of existing literature, is critically examined in this section [18].

A study by [2] examined network intrusion detection by testing RF, SVM, LSTMs, and Autoencoders using the CICIDS2017 and UNSW-NB15 datasets. The research examined model performance when applied to real-world attacks to identify a model that can best detect cyberattacks with improved accuracy and efficiency. The RF model displayed maximum accuracy at 92.3% when compared to 89.7% accuracy achieved by SVM within the ML models. LSTMs achieved 94.1% successful attack detection through their sequential approach at the expense of 200 ms computational time per task. Autoencoder achieved higher detection success than K-Means Clustering because its accuracy rate exceeded 87.8%, whereas K-Means scored at 85.4%. These results suggest that although LSTM achieves higher accuracy, it incurs higher computational costs.

A study conducted by [19] was aimed at enhancing security in satellite-terrestrial integrated networks (STIN) using four hybrid IDSs that integrate advanced feature selection with ML and DL methods. The authors optimized the STIN and UNSW-NB15 dataset feature sets using sequential forward selection based on RF to minimize computational cost while optimizing detection performance. RF-based model reached 90.5% on satellite data and 78.52% on terrestrial traffic, and DL variants incorporating RF feature extraction and Gated Recurrent Unit (GRU) reached 87% and 79%, respectively. Architectures based on Long Short-Term Memory (LSTM) and GRU have also been effective in identifying complex attack patterns, including distributed denial-of-service (DDoS) attacks. However, traditional machine learning approaches, particularly when combined with ensemble methods and multilayer perceptron models, continue to perform competitively in detecting such threats. The results demonstrate the effectiveness of hybrid IDS designs in dealing with changing threats in integrated network environments. Research has shown that machine learning methods can help to detect intrusions, especially to counter DDoS attacks.

In a study by [19], it was stated that RF with preprocessing methods like min-max scaling and outlier detection had an impressive accuracy of 99.72% at identifying DDoS attacks. In evaluation studies, several classifiers, including BayesNet, Naive Bayes, J48, Partial Decision Trees (PART), and Random Forest, have been tested on the NSL-KDD dataset alongside dimensionality reduction techniques such as Principal Component Analysis (PCA) and Random Projection (RP) [20]. The findings showed that RP combined with the PART classifier provided the best performance with a 82.0% accuracy and a false positive rate of 16.2, and the F1 score was equal in normal and anomaly classes (82.3% and 81.7%). Additionally, RP was found to be better than PCA in sustaining the classification accuracy and minimizing the computational cost, which highlights its applicability in intrusion detection systems that require efficiency and accuracy in real time.

Data balancing and feature engineering are critical in the field of cybersecurity. Techniques such as SMOTE are commonly used to address class imbalance, particularly in datasets where malicious instances are underrepresented. Flow-related features (packet size, duration, protocol type) are

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2026
DOI: https://doi.org/10.33333/lajc.vol13n2.08

always among the most important indicators of malicious activity [21]. RF is still widely used due to its feature ranking feature and ability to process high-dimensional data. In a recent study by [22], ML models (RF, SVM, Gradient Boosting), anomaly detection algorithms (K-Means, DBSCAN, Autoencoder), and DL models (Transformer, RNNs, CNNs) were compared in terms of real-time attack detection. RF was 94.5% accurate, whereas CNNs were 95.8% accurate, and their cost of computation was lower, so they were the most feasible option. The Transformers had the highest accuracy of 96.2% and required 18.3 GB of RAM and 8.5 hours for training, whereas Autoencoders had the highest true positive rate of 94% and the lowest latency of 120 ms, but the highest resource use. Although deep learning models are very accurate, they require intensive calculations and are not interpretable, which prevents their use in real-time. RF on the other hand provides the best tradeoff between accuracy, efficiency, and explainability and good generalization across datasets and low resource consumption- making it one of the top choices in the next-generation intrusion detection systems.

A significant gap in the research on lightweight model integration into real-time resource-constrained environments is evident. The gap addressed in this paper is the creation of an RF-based IDS that is optimized in terms of speed, interpretability, and efficiency. A comparison (Table I) between the existing methods shows that there are trade-offs between current methods: signature-based systems are only able to detect known threats; state-of-the-art AI models like Transformers and LSTMs are highly accurate, but impractical because of resource requirements; and methods like RP with PART classifiers have high false positive rates. This analogy supports the necessity of light, interpretable, and high-performing models like the Random Forest that can balance the detection with the feasibility of the operation.

TABLE I. LIMITATIONS OF EXISTING CYBERSECURITY APPROACHES

Author/s	Method	Detection Accuracy	Key Limitations	Operational Impact
[4]; [1]	Signature/ Rule-based	Focus on known attacks only	Restricted and limited to known attack patterns because it relies on signatures/rules.	- Increased breach Risk - High remediation costs - System downtime
[20]	Random Projection and PART Classifier	82.0% Attack Detection	Misses 18% of the attacks (breach risk), and with 16.2% it has false positives (alert fatigue).	16.2% false positives, SOC teams spend time on a false sense of security and miss real attacks.
[10]	Transforme r	96.2% Attack Detection	Heavy Resource Requirements: 18.3 GB RAM, 8.5h train time.	Cost-prohibitive to the resource-limited organisation.
[2]	LSTM	94.1% Attack Detection	The model's major drawback is its 200 ms computational time, which is a significant cost.	Extra computing costs cause more operational expenses.

The comparative analysis highlights the weaknesses of both traditional and modern approaches to cybersecurity: deep learning models, despite their high accuracy, are costly in terms of resources and cannot be applied in real-time, whereas lightweight models tend to lose their accuracy, leading to alert fatigue. Such gaps highlight the need for a solution that is correct, computationally efficient, and interpretable. To overcome this, the present study proposes an RF-based IDS that is optimized to perform real-time detection in resource-constrained environments.

IV. THE MODEL FRAMEWORK, ARCHITECTURE, AND PROPOSED METHODOLOGY

To improve cyberattack detection with high accuracy, robustness, and interpretability, this paper proposes a Random Forest-based IDS model. The framework was implemented in Python, with a Streamlit interface for model training, test data uploading, attack detection, model persistence, and result logging. The implementation also incorporates machine learning and pre-processing libraries, such as Pandas, NumPy, Scikit-learn, Imbalanced-learn, Matplotlib, and Seaborn.

A. System Architecture

The layers of architecture include the following:

- **Data Acquisition Layer**
  - Gathers unprocessed network traffic content of benchmark data: CICIDS2017, NSL-KDD, UNSW-NB15.
  - The datasets were selected due to varying traffic behaviors and attack types, allowing for the model's performance to be assessed in varying cybersecurity environments.
- **Data Preprocessing Layer**
  - Standardizes data (eliminates duplicates, blank values).
  - Compatibility between datasets was achieved by standardizing the target column across all datasets, identified dynamically using labels like label, attack cat, attack, class, and target.
  - One Hot Encoding and Min-Max Normalization were performed on categorical and numerical variables, respectively. The preprocessing tasks were implemented using: OneHotEncoder, MinMaxScaler, SimpleImputer, ColumnTransformer, and Pipeline.
  - These datasets were then split into training (70%), validation (15%), and test (15%) sets for model development and evaluation.
  - Imbalance was addressed by employing balancing techniques, specifically SMOTE, during the training process when class imbalance was detected, but not in the independent data used for evaluation.
  - Pre-processing and interpretation were handled with caution for features that were not representative of meaningful network behavior, in particular identifier-type

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2026
DOI: https://doi.org/10.33333/lajc.vol13n2.08

features, to avoid adding record-specific artefacts to the model that distracted from actual traffic-related features.

- **Feature Engineering Layer**
  - Performs feature selection using Random Forest's in-built importance ranking.
  - Computes Gini impurity to assess feature relevance.
  - The most relevant features were selected based on their importance scores to improve model efficiency, focusing on specific traffic characteristics.
  - The selected features were retained throughout the implemented workflow until approximately 95% cumulative importance had been captured.
- **Model Development Layer**
  - The Random Forest classifier was chosen as the primary detection method given its efficiency, interpretability, and robustness in dealing with high-dimensional data.
  - Uses ensemble voting by multiple decision trees.
  - Hyperparameter optimization was performed by using RandomizedSearchCV to enhance the performance of the model when training.
  - The parameter search space included the number of estimators (n_estimators = 100, 200, 300, 400, 500), maximum depth (max_depth = 15, 20, 30, 40, 50), and minimum samples required for split (min_samples_split = 2, 5, 10). The other parameters used were bootstrap=True, max_features='sqrt', criterion='gini', and random_state=42.
  - The optimization process employed n_iter=10, cv=5, scoring='accuracy' and n_jobs=4, and the best estimator found during the optimisation process was chosen as the resulting model.
- **Model Training Layer**
  - After preprocessing and feature selection, the training subset was used for model fitting.
  - The training workflow was handled in the application environment, and the progress of the models was tracked on a training status file and execution logs.
  - If class balancing was applied, only the selected training features were chosen to undergo SMOTE and random_state=42 was applied. Balanced training subset was then used for Random Forest training and hyperparameter optimization.
  - The validation subset was tracked for model behavior and the test subset was kept separate for final evaluation. This made it possible to evaluate the model on previously unseen records.
- **Evaluation Layer**
  - Evaluates model using Accuracy, Precision, Recall, and F1-Score metrics.
  - The measures were employed to assess the model's overall classification performance and the model's behavior with respect to false positive and false negative items.
  - Validation and test metrics were stored in specific result folders to ensure that the outcome would be traceable and even reusable.
- **Detection Layer**
  - The trained model was then used to detect cyberattacks on uploaded or pre-saved test data via the interface of the system.
  - Application also saved the trained model, preprocessing object, selected feature information, test metrics, validation metrics, and execution logs for re-use.
  - The design of the framework was therefore kept lightweight and deployment-friendly. But the evaluation conducted in this study was primarily based on benchmark datasets rather than real-time deployment.

B. Datasets

In this paper, three benchmark datasets are employed: the CICIDS2017, the NSL-KDD, and the UNSW-NB15. These datasets were chosen as they have a wide variety of both benign and malicious traffic characteristics, allowing the model to be tested in various intrusion detection situations. The validation of the proposed model was also improved by using three known benchmark datasets. The CICIDS2017 dataset was used because it has realistic enterprise network traffic with both benign and attack records. The NSL-KDD dataset was selected as it is still a well-known dataset for intrusion detection studies and provides several categories of attacks. The UNSW-NB15 dataset was added as it has more recent and more complex attack patterns which are relevant for the modern cybersecurity evaluation. Each data set had its target variable selected during the data pre-processing. After which the data sets were cleaned, encoded, scaled, and further reduced through feature selection using Random Forest feature importance. This enabled the selection of the most relevant variables to be used when training and evaluating the model.

C. Model Evaluation Using Benchmark Datasets

Random Forest is a technique of ensemble learning that constructs a variety of decision trees and combines their predictions to enhance generalization and combat overfitting [22]. The RF-based IDS was evaluated using three benchmark datasets: NSL-KDD (148,517 records, 31

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2026
DOI: https://doi.org/10.33333/lajc.vol13n2.08

features), CICIDS2017 (692,703 records, 37 features), and UNSW-NB15 (2,540,044 records, 49 features).

- **Mathematical Formulations:**
  - **Preprocessing**

Encoding Stage: The categorical features (such as protocol type and service) went through One-Hot Encoding.

Scanning -Min-max normalization was used to scale the numerical variables to a range of 0 to 1. The normalized value (scaled) of X represents X_{norm} Following the implementation of Min-Max scaling, represented by equation 1.

$$X_{norm} = \frac{X - X_{Min}}{X_{Max} - X_{Min}} \quad (1)$$

where:

X: The original value of a feature

X_{min}: The feature lowest value in the dataset

X_{max}: Highest feature value in the dataset

Feature Importance Formula (Gini-Based): The RF algorithm has calculated the feature importance by comparing the amount of contribution of each feature to better classification accuracy. The features that invariably resulted in larger reductions of impurity were allocated greater importance, indicating that they had a greater impact on the predictive ability. $\text{Importance}(X_m)$ as shown in Equation 2 [23].

$$Importance(X_m) = \frac{1}{B}\sum_{b=1}^{B} \Delta impurity\,(X_m, T_b) \quad (2)$$

where:

B: The total number of trees
Tb: The b-th decision tree in the forest
Δimpurity: Gini decrease from splitting X_m in tree T_b

- **Prediction**

A prediction of ŷ emerges by combining the votes of several decision trees. This ensemble approach enhanced the robustness and accuracy of the final classification, as shown in equation 3 [24]:

$$\hat{y} = mode\left(T^1(x), T^2(x), \ldots, T_n(x)\right) \quad (3)$$

where:

The $T_i(x)$ term represents the likelihood of the i-th decision tree.

- **Split Evaluation - Gini Impurity:**

RF employed Gini Impurity as its measurement tool to determine the split quality of data features. $G(p)$ is the Gini Impurity of the dataset, as presented in equation 4 [24]:

$$G(p) = 1 - \sum_{i=1}^{c} Pi^2 \quad (4)$$

where:

C = class count
p_i = likelihood of class I in the dataset p

D. *Evaluation Metrics*

This sub-section indicates that a rigorous test was conducted to evaluate the model's ability to detect attacks. A list of conventional evaluation metrics was employed to obtain a comprehensive understanding of the competence of the model.

TP = True Positives, TN = True Negatives

FP = False Positives, FN = False Negatives

The following metrics were used for evaluation:

- **Accuracy** -the percentage of cases, both true positives and true negatives, that were correctly classified out of all the instances shown by equation 5 [2].

$$\boldsymbol{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (5)$$

- **Precision and Recall** -Recall shows the number of found anomalies among all detected abnormalities, and precision calculates the proportion of these from all identified items, as shown by equations 6 and 7, respectively [2].

$$\boldsymbol{Precision} = \frac{TP}{TP+FP} \quad (6)$$

$$\boldsymbol{Recall} = \frac{TP}{TP+FN} \quad (7)$$

- **F1-Score** -The one metric that strikes a balance between precision and recall is the harmonic mean of the two, shown by equation 8 [2].

$$\boldsymbol{F1-Score} = 2 \times \frac{Precision \times Recal}{Precision + Recall} \quad (8)$$

E. *Implementation and Reproducibility Details*

The study was done in Python with a Streamlit application. Libraries used for the implementation were pandas, NumPy, joblib, matplotlib, seaborn, scikit-learn, and imbalanced-learn. The key machine learning algorithms used were Random Forest Classifier, train_test_split, RandomizedSearchCV, One Hot Encoder, Min Max Scaler, SimpleImputer, Column Transformer, Pipeline, and SMOTE. Hyperparameter optimization using RandomizedSearchCV was used to train the Random Forest model. The search space consisted of n_estimators (100, 200, 300, 400, 500), max_depth (15, 20, 30, 40, 50), min_samples_split (2, 5, 10), and other parameters like bootstrap=True, max_features='sqrt',criterion='gini', and random_state=42. The best-performing estimator for the optimization process was picked with the n_iter=10, cv=5, scoring='accuracy', and n_jobs=4 option. The application was designed to generate separate folders for data extract processing, model execution, model validation results, test results, and system logs. It also stored the trained model (rf_model.pkl), preprocessing object (preprocessor.pkl), selected feature indices (selected_features.npy), test and validation metrics, backup test data, and training-status records. All of these design decisions facilitated a smooth flow of execution, the reuse of trained artefacts, and the repeatability of the executed workflow. The data preprocessing, feature selection using Random Forest importance, balancing of training data if needed, model training, and evaluation were performed according to the experimental workflow, and the standard evaluation

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2026
DOI: https://doi.org/10.33333/lajc.vol13n2.08

criteria were used: accuracy, precision, recall, F1-score, and confusion matrix. This workflow allowed the provided results to be reported through a structured and reproducible machine learning workflow.

V. RESULTS

This section provides an overview of the empirical results obtained from applying the Random Forest-based model to three widely used benchmark cybersecurity datasets: CICIDS2017, NSL-KDD, and UNSW-NB15. Individual datasets are subjected to preprocessing and feature engineering to align with the objective of the study, which is to develop an efficient cyberattack detection model. This section aims to discuss the results of the model in differentiating between benign and malicious traffic on various datasets.

A. CICIDS2017 Dataset Results

The CICIDS2017 dataset is large in scale, containing approximately 11.8 million network flow instances and 85 features [25]. It contains both normal network traffic and various forms of malicious traffic gathered during several days, which resembles the real-life environment of an enterprise [25]. Within the framework of this study, a specific subset of the CICIDS2017 files was selected: WorkingHours.pcap_ISCX. The dataset used contained 692,703 records. It was selected because it is representative and comprises both normal and attack traffic in a real-world environment, as explained by [25]. This subset provides a balanced representation of daily network activity while reducing computational overhead.

1) Model Performance Metrics

The RF-based classifier performed remarkably well on the CICIDS2017 dataset, as can be observed in the confusion matrix shown in Table II. The confusion matrix explains the performance of the model on the test set, and it had 65,988 true negatives and 37,895 true positives. Importantly, it recorded 6 false negatives, which is equivalent to 0.026% false negatives, as well as 17 false positives, which is equivalent to 0.016% false positives. The low number of false positives and false negatives indicates strong classification performance.

TABLE II. CONFUSION MATRIX FOR THE CICIDS2017 DATASET, SHOWING TRUE/FALSE

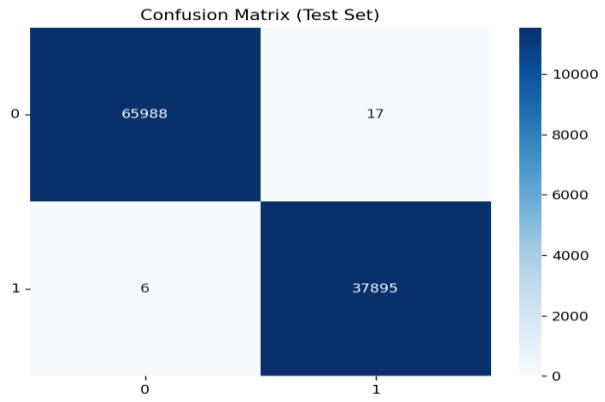


Table III below depicts the model evaluation matrices and results on the CICIDS2017 dataset. The model achieved a classification accuracy of 99.98%, accompanied by a precision of 99.96% and a recall 99.98%. The consistent effectiveness between false positives and false negatives is further supported by an F1 score of 99.97%, showing a balanced performance. The findings reveal that the model was able to successfully classify benign and malicious traffic in the tested CICIDS2017 subset.

TABLE III. RANDOM FOREST MODEL PERFORMANCE RESULTS ON CICIDS2017 DATASET

Metric	Value
Accuracy	99.98%
Precision	99.96%
Recall	99.98%
F1 Score	99.97%

Fig. 1 shows the feature importance rankings for the CICIDS2017 dataset to the RF model. The feature importance results demonstrated that the model was not only based on individual data points but also on the behavior of traffic patterns, with several traffic-related variables influencing classification.

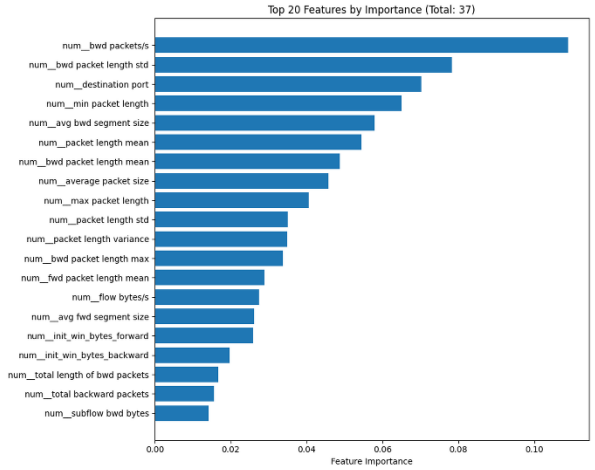


Fig. 1. Feature importance rankings for the CICIDS2017 dataset, derived from the Random Forest model.

B. NSL-KDD Dataset Results

The NSL-KDD dataset is an improved version of the original KDD dataset, KDD-99, which was specially developed to overcome the problems of duplicate and imbalanced nature that plagued the earlier one [26]. It includes four main types of attacks, namely DDoS, Probe, R2L, U2R, and normal network traffic [26]. The NSL-KDD dataset was selected as a benchmark as it contains a standard evaluation dataset for intrusion detection research and multiple categories of attacks with normal traffic. This enabled the model to be tested with well-known intrusion patterns.

1) Model Performance Metrics

For the NSL-KDD dataset, the confusion matrix revealed 11,545 true negatives and 10,708 true positives, indicating that the model successfully classified most of the normal and malicious samples. The model generated 14 false positives and 11 false negatives, indicating strong detection performance.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2026
DOI: https://doi.org/10.33333/lajc.vol13n2.08

TABLE IV. CONFUSION MATRIX FOR THE NSL-KDD DATASET, SHOWING TRUE/FALSE POSITIVES/NEGATIVES

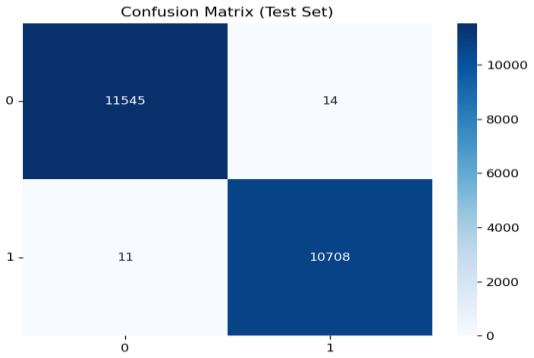


Table V. below depicts the model evaluation matrices and results on the NSL-KDD dataset. The results of the evaluation were a classification accuracy of 99.89%, with a precision of 99.87%, going just a bit higher than the recall of 99.90%. This correlation has produced a balanced F1 score of 99.88%, meaning that both attack detection and false alarm prevention have been performed consistently. The values are preserved throughout the reported results, and it can be seen that they are an unambiguous measure of model performance in the NSL-KDD dataset.

TABLE V. RANDOM FOREST MODEL PERFORMANCE RESULTS ON NSL-KDD DATASET

Metric	Value
Accuracy	99.89%
Precision	99.87%
Recall	99.90%
F1 Score	99.88%

Fig. 2 shows the feature importance for the NSL-KDD dataset.

The feature importance analysis revealed that the traffic-related rate, num_src_bytes, and num_dst_bytes are important for classification. These features are true traffic behavior features and so are applicable in intrusion detection.

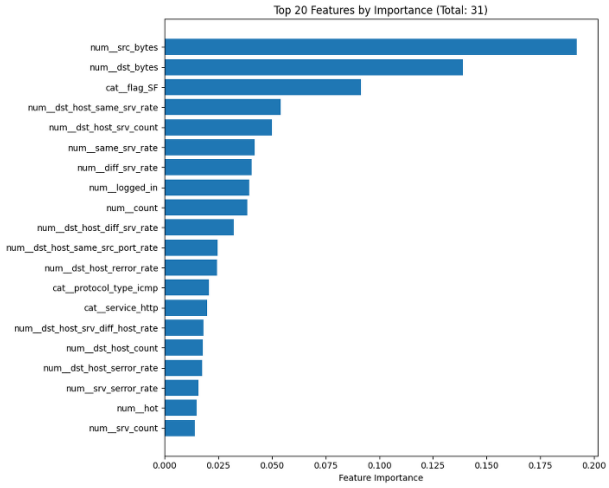


Fig. 2. *Feature importance rankings for the NSL-KDD dataset, derived from the Random Forest model*

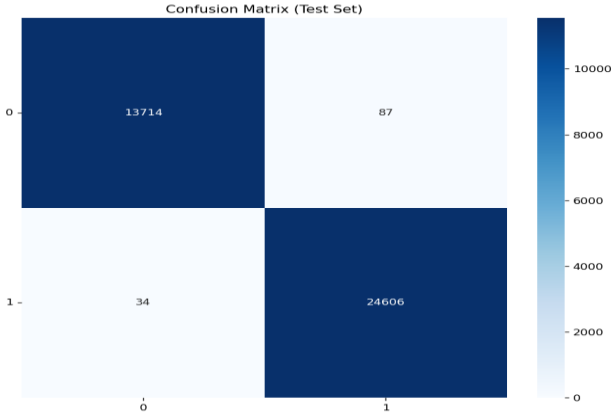
C. UNSW NB15 Dataset Results

The UNSW-NB15 dataset offers recent network traffic, including state-of-the-art types of attacks, meaning Exploits, Fuzzers, and Shellcode [27]. The UNSW-NB15 was generated with IXIA Perfect Storm, a professional network traffic generator, which exhibits high variability, thus making it a strenuous and realistic benchmark to test IDSs [28]. The entire dataset of 257,673 records was utilized for analysis, and 33 features were used.

1) *Model Performance Matrices*

The confusion matrix of the model on the test set, depicted in Table VI, gives a quantitative measure of the performance of the model concerning its classification. The confusion matrix revealed 13,714 true negatives and 24,606 true positives, highlighting good classification accuracy for both benign and malicious traffic. The model achieved a low number of 87 false positives and a low number of 34 false negatives, among the correctly classified instances.

TABLE VI. CONFUSION MATRIX FOR THE UNSW NB15, SHOWING TRUE/FALSE POSITIVES/NEGATIVES



The RF model provided good detection performance in the modern UNSW-NB15 dataset, as depicted in Table VII. below. The overall accuracy of the classifier was 99.69%, the precision was 99.65%, the recall was 99.86%, and the F1 score of 99.75%. The values are slightly lower than those found in CICIDS2017 and NSL-KDD, as attack patterns are far more complex and varied in the UNSW-NB15 dataset.

TABLE VII. RANDOM FOREST MODEL PERFORMANCE RESULTS ON UNSW NB15 DATASET

Metric	Value
Accuracy	99.69%
Precision	99.65%
Recall	99.86%
F1 Score	99.75%

Fig. 3 shows the feature importance rankings for the UNSW-NB15 dataset. The feature importance analysis showed that the traffic related attributes namely sttl and ct_state_ttl were significant features affecting the classification performance. Identifier-based variables, like num_id, were however interpreted with a certain caution as they do not describe intrinsic network behaviour and might be misinterpreted as an indexing effect and not as a significant attack characteristic.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2026
DOI: https://doi.org/10.33333/lajc.vol13n2.08

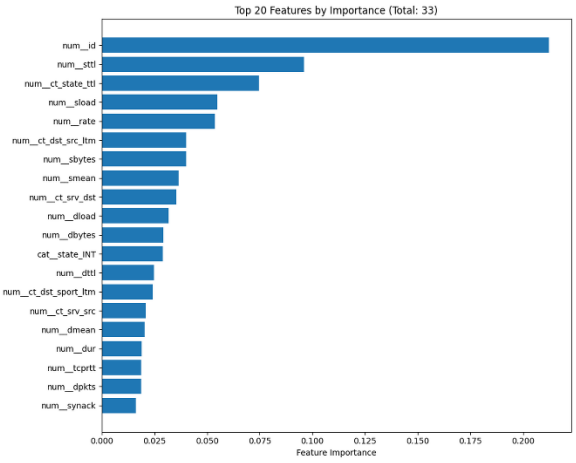


Fig. 3. *Feature importance rankings for the UNSWNB15 dataset, derived from the Random Forest model*

D. Summary of Results Across Datasets

In all three datasets, the model based on the Random Forest algorithm showed consistently high classification performance. CICIDS2017, NSL-KDD, and UNSW-NB15 were found to be the top three performing datasets. The slightly lower performance in the UNSW-NB15 data set is due to the fact that the patterns are more complex and newer. In all the datasets, the confusion matrices exhibited low false positive and false negative rates, demonstrating that the model could maintain a balance between detecting malicious activity and giving few false alarms in the benchmark set used. Overall, the results show that the Random Forest model is effective for cyberattack detection within the experimental setup used in this study. The results discussed here are in line with its suitability to be a lightweight and practical detection approach, but do not fully represent a real-time, fully operational deployment.

VI. DISCUSSION

The findings of this study demonstrate that the Random Forest model can be effectively used for cyberattack detection in a variety of datasets. Its high generalizability and reliability are reflected in accuracy, precision, and recall values exceeding 99%. The minor difference in the performance of the datasets, especially the reduced accuracy of the UNSW-NB15 dataset when compared to CICIDS2017 and NSL-KDD, is due to the complexity and stealthiness of the attacks in the UNSW-NB15 dataset. This is consistent with literature results that current data sets with sophisticated types of attacks are more difficult to detect using detection systems.

On the CICIDS2017 dataset, the current model achieved an accuracy of 99.98%, precision of 99.96%, recall of 99.98%, and an F1-score of 99.97%, based on 692,703 records and 37 features. Comparatively, a study by [27] achieved an equally good performance of 99.94% accuracy, 99.94% precision and recall, and 99.94% F1-score with only 10 features. A study by [2] shows inferior results on every measure (92.3% accuracy, 90.5% precision, 88.2% recall, and 89.3% F1-score), and the study employed 80 features.

The same trend can be observed in the analysis of the UNSW-NB15 dataset. A subset of 257,673 records and 33 features was used in the current paper, and the detection accuracy is 99.86%, 99.65% precision, 99.86% recall, and a 99.75% F1-score. A study by [29] was conducted based on the full dataset containing 2,540,044 records, whereas the present one was based on a sample of the subset of 257,673 records. The dataset size was not indicated by [30], while the study conducted by [31] used a subset of 257,673, which aligns with the dataset used in the current study. The number of selected features ranged from 19 in [31] to 49 in [29], while the current study used 33 features. As the performance matrices indicate, study by [31] achieved an accuracy of 95.05%, which is the lowest in the comparison. The results reported in [29] were significantly high and comprised 99.42% accuracy, 99.71% precision, 99.63% recall, and 99.67% F1-score. The current model achieved comparable performance to the studies reviewed with 99.86 % accuracy, 99.65 % precision, 99.86 % recall, and 99.75 % F1-score. The study by [30] achieved 98.7% accuracy, but did not provide other performance indicators, which is why the comparison cannot be considered complete.

On the NSL-KDD dataset, the current model showed once again superior performance with 99.89% accuracy, 99.87% precision, 99.90% recall, and 99.88% F1-score on the full dataset of 148,517 records and 31 features. The matrices of evaluation indicate that [19] achieved a high accuracy of 99.72%, an F1-score of 99.70%, the highest precision of 99.84%, and a recall of 99.56%. The current study has been found to compete effectively with the highest accuracy of 99.89%, precision of 99.87%, recall of 99.90%, and an F1-score of 99.88%. In a study by [32] had a bit lower performance in all matrices (98.75-98.76%), although the number of features used by them was the lowest. Study by [30] did not present other performance measures for comparison with the high accuracy of 99.65%. Overall, the comparative analysis highlights the strength of the existing model in a wide range of datasets and experimental states. Simultaneously, it shows significant discrepancies in the manner the studies report dataset size, feature selection, and evaluation metrics. Such discrepancies make comparisons difficult and underline the necessity of developing the standardized evaluation frameworks within the research of intrusion detection. These frameworks would help in reproducibility, transparency, and more credible benchmarking of future models.

VII. CONCLUSION

This work presented and tested a Random Forest-based Intrusion Detection System (IDS) for the detection of cyberattacks using three benchmark datasets, namely CICIDS2017, NSL-KDD, and UNSW-NB15. The model was created to be lightweight and efficient for distinguishing network traffic as benign or malicious. The results demonstrate that the Random Forest model achieves high classification accuracy across different datasets in various traffic conditions. The evaluation results showed that the model performance was balanced in terms of accuracy, precision, recall, and F1-score, and that the number of false positives as well as the number of false negatives in the evaluated data sets were kept at a low level. These results show how well the model can differentiate between normal and attack traffic in various circumstances. It was also noted that the model was consistent across the various datasets, with slightly poorer performance on the UNSW-NB15 dataset, as

the attacks were more complex and varied. This indicates that dataset complexity plays an important role in the intrusion detection performance of the model. The feature analysis revealed that traffic-related features provided more relevant information for classification than identifier-based features. These findings highlight the importance of using network-related features that are relevant to actual network behavior for the development of intrusion detection systems. The study shows the effectiveness and interpretability of the Random Forest algorithm for cyberattack detection within the scope of the experimental evaluation carried out. Furthermore, the application of a structured preprocessing and evaluation pipeline and multi-dataset validation results in robust findings. The model was designed for deployment-oriented attributes, while the evaluation in this study was conducted in a benchmark-based environment. The results, therefore, do not necessarily reflect the capability for detection in a fully validated real-time operation, but rather under controlled experimental conditions. This study does not directly compare with deep learning models like LSTM and RNN under the same experimental conditions. For this reason, it is not claimed that the above models are superior. Rather, the results indicate that Random Forest-based approach is a practical solution in terms of performance, interpretability, and computational requirements for intrusion detection problems. Future research should be conducted to test the model in a fully operational setting, measuring inference latency, throughput, memory consumption and processing capacity in real-time. Moreover, future studies can investigate the use of hybrid models using a combination of Random Forest and additional machine learning or deep learning models for more accurate detection of sophisticated and evolving cyberattacks.

REFERENCES

[1] F. Tao, M. Akhtar, and Z. Jiayuan, "The future of Artificial Intelligence in Cybersecurity: A Comprehensive Survey," EAI Endorsed Trans. Creat. Technol., vol. 8, no. 28, p. 170285, Aug. 2021, doi: 10.4108/eai.7-7-2021.170285.

[2] V. Jain and A. Mitra, "Real-Time Threat Detection in Cybersecurity: Leveraging Machine Learning Algorithms for Enhanced Anomaly Detection," in Advances in Computational Intelligence and Robotics, M. A. Almaiah and Y. Maleh, Eds., IGI Global, 2024, pp. 315–344. doi: 10.4018/979-8-3693-7540-2.ch014.

[3] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, "A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions," Electronics, vol. 12, no. 6, p. 1333, Mar. 2023, doi: 10.3390/electronics12061333.

[4] M. Markevych and M. Dawson, "A Review of Enhancing Intrusion Detection Systems for Cybersecurity Using Artificial Intelligence (AI)," Int. Conf. Knowl.-BASED Organ., vol. 29, no. 3, pp. 30–37, June 2023, doi: 10.2478/kbo-2023-0072.

[5] Seacom, "Financial impact of security breaches is highest in SA," Jan. 04, 2024. [Online]. Available: https://seacom.co.za/news/financial-impact-of-security-breaches-is-highest-in-sa

[6] Aswa, "AI-Powered Cybersecurity: Leveraging Deep Learning for RealTime Threat Detection and Prevention," 2025, 2025, doi: 10.18535/ijecs/v14i01.4975.

[7] Ayodeji Oyindamola Ikudabo, Chinedu C. Onyeje, Daniel O. T. Ihenacho, and K. C. Nwafor, "Mitigating cybersecurity risks in financial institutions: The role of AI and data analytics," Int. J. Sci. Res. Arch., vol. 13, no. 1, pp. 2895–2910, Oct. 2024, doi: 10.30574/ijsra.2024.13.1.2014.

[8] A. Khemka, "The impact of cyber attacks on financial institutions and the need for improved security measures.," vol. 9, no. 10, 2024.

[9] Md Anwarul Matin Jony, Rashedul Islam, and F. H. Muhammad Saqib Jalil, "AI-Powered Cybersecurity in Financial Institutions: Enhancing Resilience Against Emerging Digital Threats," Adv. Int. J. Multidiscip. Res., vol. 2, no. 6, p. 1113, Nov. 2024, doi: 10.62127/aijmr.2024.v02i06.1113.

[10] H. Hussain, M. Kainat, M. Tunio, and T. Ali, "Leveraging AI and Machine Learning to Detect and Prevent Cyber Security Threats," Jan. 2025, doi: 10.5281/ZENODO.14714679.

[11] N. Katiyar, S. Tripathi, Mr. P. Kumar, Mr. S. Verma, A. K. Sahu, and S. Saxena, "AI and Cyber-Security: Enhancing threat detection and response with machine learning.," Educ. Adm. Theory Pract., Apr. 2024, doi: 10.53555/kuey.v30i4.2377.

[12] M. Mohammed, A. J. Mohammed, U. U. M. Mohammed, and Z. A. Mohammed, "Advancements in AI-Based Security and Threat Detection," IJARCCE, vol. 13, no. 4, Mar. 2024, doi: 10.17148/IJARCCE.2024.13459.

[13] R. C. D. Oliveira and R. D. D. S. E. Silva, "Artificial Intelligence in Agriculture: Benefits, Challenges, and Trends," Appl. Sci., vol. 13, no. 13, p. 7405, June 2023, doi: 10.3390/app13137405.

[14] A. AI-Kuwaiti et al., "A Review of the Role of Artificial Intelligence in Healthcare," J. Pers. Med., vol. 13, no. 6, p. 951, June 2023, doi: 10.3390/jpm13060951.

[15] C. Meirinhos, L. Fernandes, and M. Meirinhos, "The emergence of Artificial Intelligence in Education," 2023.

[16] K. Meduri, H. Gonaygunt, and G. S. Nadella, "Evaluating the Effectiveness of AI-Driven Frameworks in Predicting and Preventing Cyber Attacks," Int. J. Res. Publ. Rev., vol. 5, no. 3, pp. 6591–6595, Mar. 2024, doi: 10.55248/gengpi.5.0324.0875.

[17] R. Kaur, D. Gabrijelčič, and T. Klobučar, "Artificial intelligence for cybersecurity: Literature review and future research directions," Inf. Fusion, vol. 97, p. 101804, Sept. 2023, doi: 10.1016/j.inffus.2023.101804.

[18] A. O. Adewusi, U. I. Okoli, T. Olorunsogo, E. Adaga, D. O. Daraojimba, and O. C. Obi, "Artificial intelligence in cybersecurity: Protecting national infrastructure - A USA review," World Journal of Advanced Research and Reviews, vol. 21, no. 1, pp. 2263–2275, 2024, doi: 10.30574/wjarr.2024.21.1.0313

[19] I. Avci and M. Koca, "Cybersecurity Attack Detection Model, Using Machine Learning Techniques," Acta Polytech. Hung., vol. 20, no. 7, pp. 29–44, 2023, doi: 10.12700/APH.20.7.2023.7.2.

[20] F. Nabi and X. Zhou, "Enhancing intrusion detection systems through dimensionality reduction: A comparative study of machine learning techniques for cyber security," Cyber Secur. Appl., vol. 2, p. 100033, 2024, doi: 10.1016/j.csa.2023.100033.

[21] M. Luay, S. Layeghy, S. Hosseininoorbin, M. Sarhan, N. Moustafa, and M. Portmann, "Temporal Analysis of NetFlow Datasets for Network Intrusion Detection Systems," Mar. 09, 2025, arXiv: arXiv:2503.04404. doi: 10.48550/arXiv.2503.04404.

[22] H. A. Salman, A. Kalakech, and A. Steiti, "Random Forest Algorithm Overview," Babylon. J. Mach. Learn., vol. 2024, pp. 69–79, June 2024, doi: 10.58496/BJML/2024/007.

[23] D. Koutsandreas and I. Keppo, "Harnessing machine learning algorithms to unveil energy efficiency investment archetypes," Energy Rep., vol. 12, pp. 3180–3195, Dec. 2024, doi: 10.1016/j.egyr.2024.09.009.

[24] Putta Srivani, "Integrating Natural Language Processing with AdaBoost, Random Forest, and Logistic Regression for an Advanced Ensemble-Based Network Intrusion Detection Model," J. Inf. Syst. Eng. Manag., vol. 10, no. 3s, pp. 264–283, Jan. 2025, doi: 10.52783/jisem.v10i3s.386.

[25] Zafar Iqbal Khan, Mohammad Mazhar Afzal, and Khurram Naim Shamsi, "A Comprehensive Study on CIC-IDS2017 Dataset for Intrusion Detection Systems," Int. Res. J. Adv. Eng. Hub IRJAEH, vol. 2, no. 02, pp. 254–260, Feb. 2024, doi: 10.47392/IRJAEH.2024.0041.

[26] Y. Sahli, "comparison of the NSL-KDD dataset and its predecessor the KDD Cup '99 dataset," Int. J. Sci. Res. Manag., vol. 10, no. 04, pp. 832–839, Apr. 2022, doi: 10.18535/ijsrm/v10i4.ec05.

[27] Md. A. Talukder et al., "Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction," J. Big Data, vol. 11, no. 1, p. 33, Feb. 2024, doi: 10.1186/s40537-024-00886-w.

[28] S. M. Kasongo, "An Advanced Intrusion Detection System for IIoT Based on GA and Tree Based Algorithms," IEEE Access, vol. 9, pp. 113199–113212, 2021, doi: 10.1109/ACCESS.2021.3104113.

[29] S. More, M. Idrissi, H. Mahmoud, and A. T. Asyhari, "Enhanced Intrusion Detection Systems Performance with UNSW-NB15 Data Analysis," Algorithms, vol. 17, no. 2, p. 64, Feb. 2024, doi: 10.3390/a17020064.

[30] J. Note and M. Ali, "Comparative Analysis of Intrusion Detection System Using Machine Learning and Deep Learning Algorithms," Ann. Emerg. Technol. Comput., vol. 6, no. 3, pp. 19–36, July 2022, doi: 10.33166/AETiC.2022.03.003.

[31] I. H. Putro, "Evaluating the Performance of Machine Learning Classifiers for Network Intrusion Detection: A Comparative Study Using the UNSW-NB15 Dataset," Teknika, vol. 14, no. 2, pp. 330–338, July 2025, doi: 10.34148/teknika.v14i2.1276.

[32] A. M. A. Abdullah and K. A. Abood, "Comparative Analysis of Machine Learning Techniques for Intrusion Detection in IoT Networks," Univ. Aden J. Nat. Appl. Sci., vol. 28, no. 2, pp. 53–60, Apr. 2025, doi: 10.47372/uajnas.2024.n2.a05.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2026
DOI: https://doi.org/10.33333/lajc.vol13n2.08

AUTHORS

Phathutshedzo Cyprin Ramuhovhi

Phathutshedzo Ramuhovhi is an IT professional and researcher with a strong background in information technology and applied artificial intelligence. His academic training has focused on cybersecurity, machine learning, and data-driven systems, with a particular interest in the detection and prevention of cyberattacks. He completed his Master's research in intrusion detection systems, where his work explored the use of machine learning models for cyberattack detection across multiple benchmark datasets.

He is currently advancing his research in artificial intelligence and cybersecurity, with a focus on federated learning, explainable artificial intelligence, and intelligent intrusion detection systems. His work integrates machine learning techniques with practical cybersecurity applications to improve detection accuracy and system efficiency.

In addition to his academic work, he serves as a Maintenance & Support Delivery Manager, contributing to IT operations and system support. His expertise in Python programming, data analysis, and machine learning model development reflects a strong integration of theoretical knowledge and practical industry experience.

Naume Sonhera

Dr Naume Sonhera is a Senior Lecturer and Head of the Department of Computer Science within the Faculty of Applied and Computer Sciences at the Vaal University of Technology (VUT), South Africa. She holds a Doctor of Philosophy (PhD) in Information Systems, a Master of Science (MSc) in Computer Science, and a Bachelor of Science in Education (Licentiate Degree) in Mathematics.

She has held various academic and leadership roles throughout her career, including ICT Coordinator, Head of Academics, and Information Technology Manager, and has also served as Acting Campus Principal at one of the university's satellite campuses.

Dr Sonhera is an established researcher with a strong record of impactful research outputs in Computer Science and ICT. Her research interests include Information and Communication Technologies for Development (ICT4D), cloud computing, cybersecurity, cyber threats, cyberbullying, and artificial intelligence.

She is professionally affiliated with the Institute of Information Technology Professionals South Africa (IITPSA) and the South African Institute of Computer Scientists and Information Technologists (SAICSIT), and has received recognition for her contributions to teaching excellence.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2026

AUTHORS

Tranos Zuva

Professor Tranos Zuva is a Professor of Computer Science at the Vaal University of Technology (VUT) and serves as the MICT SETA Fourth Industrial Revolution (4IR) Research Chair. He has over 30 years of experience in teaching, research, innovation, and academic leadership, and is widely recognized for his contributions in Artificial Intelligence, Cybersecurity, Data Science, Software Engineering, Digital Transformation, and Emerging Technologies.

He has published more than 200 peer-reviewed journal articles and conference papers and has successfully supervised numerous Master's and Doctoral students. His work has received significant recognition for its impact on both academia and industry.

Professor Zuva actively promotes industry–academic collaboration, innovation, and capacity building. He plays a leading role in advancing 4IR initiatives, digital skills development, and the application of technology-driven solutions to address societal and industrial challenges in South Africa and beyond.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2026