UDC 004.77:004.8 **doi:** https://doi.org/10.20998/2522-9052.2026.1.13

Akwaeno Isong[1], Bliss Utibe-Abasi Stephen[1], Philip Asuquo[1], Chijioke Ihemereze[2], Imoh Enang[2]

[1] Department of Computer Engineering, University of Uyo, Uyo, Nigeria
[2] Department of Computer Engineering, Federal Polytechnic Nekede, Owerri, Nigeria

MACHINE LEARNING BASED CLOUD COMPUTING INTRUSION DETECTION

Abstract. Based on today's technologically networked world**,** a sophisticated networking technology known as Software-Defined Networking (SDN) is utilized in cloud computing environments to improve the effectiveness of network management. However, SDN's centralized nature makes it vulnerable to DDoS attacks**.** This study introduces a technique for detecting DDoS attacks within a cloud computing setting. The research seeks to apply an ensemble machine learning approach for statistically identifying DDoS attacks in cloud network traffic, categorizing them as either harmful or harmless. Various machine learning algorithms, including K-Nearest Neighbors, Random Forest (RF), and Decision Tree, were utilized as foundational classifiers in the suggested ensemble machine learning model. A dataset of SDN–DDoS attacks was utilized to assess the efficacy of the base classifiers. The classifiers were trained using 80% of the dataset and evaluated on 20%. The results of the experiment indicated that the Random Forest and Random Forest classifiers attained 100% accuracy, whereas the K-Nearest Neighbor classifier achieved an accuracy of 98.21%. The ensemble machine learning model employed a majority voting technique for final prediction and achieved an accuracy of 100% on the test set, ranking as the best compared to benchmark models**.**

Keywords: Cloud Computing; Attack Classification; Machine Learning; Threat Detection; IaaS; PaaS; SaaS, Intrusion Detection System; Artificial Intelligence; Deep Learning; Feature Selection; Classification Algorithms; Anomaly Detection.

Introduction

An Intrusion Detection System (IDS) serves as a vital security tool aimed at detecting and addressing different cyber threats. These systems are essential for detecting suspicious activities at both the host and network levels. Recently, Machine Learning (ML) methods have greatly improved IDS performance, offering high precision and efficient identification of new cyber-attacks.

The primary research challenge tackled in this article involves creating a dependable and versatile intrusion detection algorithm designed specifically for identifying Distributed Denial of Service (DDoS) attacks within Software-Defined Networking (SDN) environments. SDN is gaining traction in contemporary networking infrastructures owing to its centralized control and programmability, which offer significant flexibility and scalability. Nevertheless, this centralization renders SDN especially susceptible to focused DDoS attacks, capable of incapacitating the entire network by inundating the SDN controller.

Conventional Intrusion Detection Systems (IDS) frequently find it challenging to adapt to the distinct features and fluid nature of Software-Defined Networking (SDN) environments. Current methodologies may fall short in effectively tackling the particular difficulties presented by Distributed Denial of Service (DDoS) attacks within SDN, including the capacity to identify and counteract these threats while minimizing false positives in real-time.

Given the critical role of SDN in cloud computing and other modern network infrastructures, there is an urgent need for advanced detection methods that can reliably and accurately identify DDoS attacks within SDN. This research focuses on leveraging the SDN-DDoS attack dataset to develop and evaluate a machine learning-based Network-Based Intrusion Detection System (NIDS) that is specifically optimized for SDN environments. The proposed system aims to enhance detection accuracy, reduce false alarms, and improve the overall security posture of SDN networks against DDoS attacks.

1. Literature Review

Recently, the significance of intrusion detection has escalated due to the increasing prevalence of cyberattacks [1]. Various methods have been employed by researchers in experiments to provide solutions to issues related to cyberattacks. One notable method is the stacked ensemble learning technique discussed in [2], which utilized gradient boosting and Random Forest as foundational classifiers, resulting in an accuracy of 91.06% when assessed using the NSL-KDD dataset. Additionally, another method presented an optimal Support Vector Machine (OSVM) for Intrusion Detection Systems (IDS) in Wireless Sensor Networks (WSN), achieving an accuracy of 94.09% and a detection rate of 95.02% when tested on the NSL KDDC up 99 dataset [3].

In [4], the researchers sought to minimize both false-positive and false-negative rates in intrusion detection systems specifically designed for web-based attacks. The experimental findings indicated that, out of the three algorithms tested, the J48 decision tree algorithm yielded the highest True Positive rate (94.5%), 94.7% of Precision, and 94.5% Recall rate when assessed on the meticulously refined CSIC 2010 HTTP dataset. An evaluation of twelve machine learning algorithms (Logistic Regression, Naïve Bayes, K-Nearest Neighbour (KNN), Decision Tree (DT), AdaBoost, Random Forest, Convolutional Neural Network, CNN-LSTM, LSTM, GRU, Simple RNN, and DNN) was carried out in [5].This evaluation utilized three publicly accessible datasets: CICIDS-2017, UNSW-NB15, and the Industrial Control System (ICS) cyberattack datasets. The results of this evaluation confirm that the Random Forest (RF) algorithm demonstrates superior performance in terms of accuracy, precision, recall, F1-score, and Receiver Operating Characteristic (ROC) curves across all datasets analyzed.

In reference [6], the authors developed a MultiTree algorithm that utilized a decision tree,kNN, random forest, and DNN as foundational classifiers to create an ensemble adaptive voting algorithm, which achieved an accuracy of 85.2% when tested on the NSL-KDD dataset.

The technique presented in [7] is constructed using a DT classifier, a RF classifier, and support vector machines, applying recursive feature elimination (RFE) technique to remove irrelevant features from the benchmark dataset, NSL-KDD. The results indicated that the Random Forest algorithm performed optimally with the chosen features for intrusion detection systems (IDS). To minimize the misclassification rate in detecting DDoS attacks, [8] utilized Mutual Information (MI) and Random Forest Feature Importance (RFFI) methods to identify the most pertinent features, which were then applied to Random Forest (RF), Gradient Boosting (GB), Weighted Voting Ensemble (WVE), K-Nearest Neighbour (KNN), and Logistic Regression (LR). The overall prediction accuracy of RF with 16 features reached 0.99993, and with 19 features, it improved to 0.999977, outperforming other methodologies.

In [9], the authors proposed an innovative intrusion detection system that integrates a fuzzy c-means clustering (FCM) algorithm with a support vector machine (SVM) to enhance the accuracy of anomaly detection within a cloud computing environment, achieving a relatively low false alarm rate compared to existing methods. Through performance evaluation and comparative analysis, the proposed approach attained a false negative rate of 0.003%, an accuracy of 97.37%, and a true positive rate of 97.90%

A hybrid intrusion detection system was presented in [10], which integrates SVM and genetic algorithm (GA) methodologies, complemented by a novel fitness function designed to assess system accuracy. This system was tested on two datasets, CIDS2017 and KDDCUP99, achieving a remarkable accuracy rate of 99.3%, surpassing previous benchmarked studies.

In [11], an ensemble-based machine learning strategy was employed, utilizing four classifiers—Boosted Tree, Bagged Tree, Subspace Discriminant, and RUSBoost—along with a voting mechanism to create an intrusion detection model, which was assessed on the CICIDS2017 dataset.

The results indicated an improved accuracy of 97.24% with a reduced number of false alarms compared to leading-edge methodologies.

An intrusion detection algorithm based on an ensemble support vector machine with bag representation is established in [12]. The bag representation aggregates the related samples into a bag, which can be represented as a feature matrix. Experimental findings reveal that intrusion detection utilizing bag representation yields superior precision and recall rates for ongoing attacks compared to individual data flows. A framework to assess the performance of Random Forest and XGBoost in classifying and predicting DDoS attack types was proposed in [13]. The evaluation on the UNWS-NP-15 dataset showed that Random Forest and XGBoost achieved an average accuracy of 89% and 90%, respectively.

The authors in [14] adopted a hybrid methodology by integrating k-means with the RF algorithm for binary classification, alongside CNN, LSTM, and various other deep learning techniques to further categorize abnormal events into distinct attack types. The experimental outcomes indicate that the proposed model exhibits superior true positive rates (TPR) for the majority of attack events, enhanced data pre-processing speed, and potentially reduced training duration.

Navigating through the sphere of Software-Defined Networking (SDN), the authors referenced in [15] assessed several significant feature selection techniques for machine learning in the context of DDoS detection. The findings indicate that the RF classifier is capable of training a model with an impressive accuracy of 99.97% when utilizing feature subsets selected through the Recursive Feature Elimination (RFE) method.

In [16], novel features pertinent to DDoS attacks were identified and recorded in a CSV file to construct the dataset. A hybrid machine learning model that integrates a Support Vector Classifier with Random Forest was employed for classification, resulting in a testing accuracy of 98.8% alongside a notably low false alarm rate.

A deep learning approach is explored in [17], which utilizes a CNN to identify various attacks within a Software-Defined Network (SDN). The results of the experiment demonstrate that the proposed model achieves a remarkable 100% accuracy, with a minimal degradation rate of 2.3% in throughput and 1.8% in latency when implemented in a larger system.

Conversely, in [18], a detection model based on CNN is introduced with the aim of tackling the challenges posed by DDoS attacks. The evaluation of this experiment focused on accuracy, sensitivity, and specificity, yielding results of 99.72%, 99.69%, and 99.71%, respectively.

In [19], a linear SVN model is trained using a kernel radial basis function on features extracted from traffic flow data and statistics. Various algorithm, including Naive Bayes, KNN, DT and RF, were employed and compared against the SVM model to enhance detection performance. The experimental outcomes confirm that the system effectively identifies attacks with a low rate of false alarms and high accuracy in comparison to other related methodologies. Additionally, an ensemble machine learning technique is implemented in [14], utilizing K-means++ for the grouping of training data and Random Forest as the foundational classifier, achieving a detection accuracy of 100%.

2. Purpose and Objectives of the study

The aim of the article istodevelop a machine learning-based cloud computing intrusion detection. An Intrusion Detection System (IDS) is a critical security mechanism designed to identify and mitigate various cyber threats. These systems are essential for detecting suspicious activities at both the host and network levels. In recent years, the application of ML techniques has greatly improved the efficacy of IDS, offering high accuracy and effective identification of new cyber-attacks.

The research challenge addressed in this article is the development of a reliable and adaptable intrusion

detection algorithm specifically tailored for detecting Distributed Denial of Service (DDoS) attacks in Software-Defined Networking (SDN) environments. SDN, with its centralized control and programmability, is increasingly adopted in modern networking infrastructures due to its flexibility and scalability. However, this centralization also makes SDN particularly vulnerable to targeted DDoS attacks, which can disrupt the entire network by overwhelming the SDN controller.

Traditional Intrusion Detection System (IDS) solutions frequently encounter difficulties in adapting to the distinct characteristics and dynamic nature of Software-Defined Networking (SDN) environments. Current methodologies may fall short in effectively tackling the specific challenges presented by Distributed Denial of Service (DDoS) attacks within SDN, particularly in terms of real-time detection and mitigation while minimizing false positives.

Given the critical role of SDN in cloud computing and other modern network infrastructures, there is an urgent need for advanced detection methods that can reliably and accurately identify DDoS attacks within SDN. This study concentrates on utilizing the SDN-DDoS attack dataset to create and assess a ML-driven Network-Based Intrusion Detection System (NIDS) that is tailored specifically for SDN environments. The proposed system aims to enhance detection accuracy, reduce false alarms, and improve the overall security posture of SDN networks against DDoS attacks.

3. Research materials and methods

The work presented employs ML methods, such as KNN, SVM, and RF. This research concentrates on the subsequent attacks in DDoS:

1. TCP-SYN Flood Attack.
2. UDP Flood Attack.
3. ICMP Flood Attack.

3.1 Machine Learning (ML). This represents a rapidly advancing methodology for forecasting and mitigating security risks and threats. ML is a branch of Artificial Intelligence dedicated to creating computational frameworks and statistical models derived from existing datasets, commonly known as "Training Data" [20].

The methodology employed in this study involves utilizing ML techniques to identify DDoS attacks in SDN. The dataset used for both training and testing the algorithms is the SDN-DDoS (ICMP, TCP, UDP) attack dataset. Preprocessing steps have been carried out on the dataset, including face selection, label encoding, and data normalization. This data has been split into training and testing sets to train and evaluate the ML algorithms in the model. The machine learning models utilized in this study for identifying DDoS attacks include:

1. K-Nearest Neighbor (KNN).
2. Random Forest.
3. Decision Tree.

3.2 Material and Method. This discusses the steps of the methodology for developing a machine learning-based cloud computing intrusion detection system (Fig. 1). The proposed method involves the following main steps:

1. **Dataset Selection:** Choosing the appropriate dataset for utilization.
2. **Selection of Tools and Language:** Identifying the tools and programming languages used for implementation.
3. **Data Preprocessing:** Utilizing methods to manage extraneous data. Data standardization and scaling were conducted using the Standard Scaler from Scikit-Learn.
4. **Application of Machine Learning Techniques:** Implementing ML models to classify attacks.
5. **Data Splitting:** Segmenting the dataset into training and testing subsets. During this phase, the proposed model is constructed and trained.
6. **Model Evaluation:** Evaluating the efficacy of the model on the SDN-DDoS attack dataset.

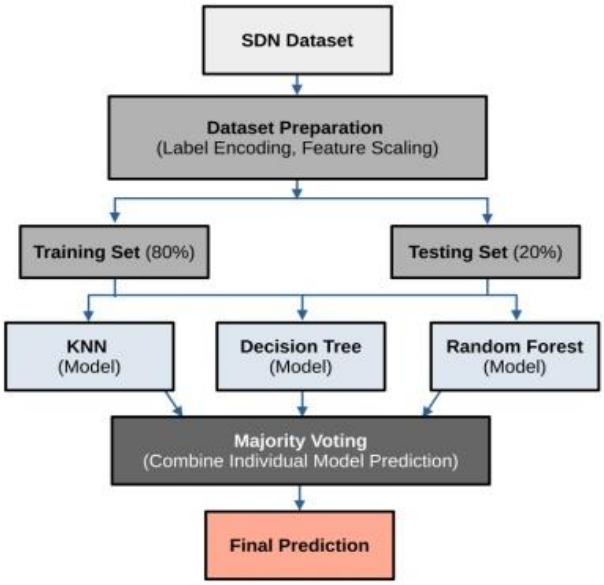

Fig. 1. Systematic diagram for the implementation of Machine learning-based Cloud computing intrusion detection

3.3 Dataset. According to this research paper, we utilized a highly reputable and extensively curated dataset from the Digital Commons Data Repository, widely recognized for the integrity and authenticity of the data deposited. The dataset selected for this study consists of DDoS attacks in SDN, including ICMP, TCP, and UDP floods (Fig. 2). It has been rigorously vetted and widely referenced in various scholarly publications [21].

Digital Commons Data serves as an institutional repository for researchers, administrators, and data curators to store, manage, publish, and preserve research datasets. Researchers worldwide depend on this repository due to its strict data collection and validation protocols, along with the open access it offers to the scientific community, thereby ensuring transparency and reproducibility in research. Digital Commons Data is a turnkey, cloud-hosted, and fully supported module that delivers all the necessary functionality to achieve an institutional research data management program without additional technical investment. All software maintenance, configuration, and implementation are managed by Elsevier teams, saving users valuable time and reducing the need for local IT support.

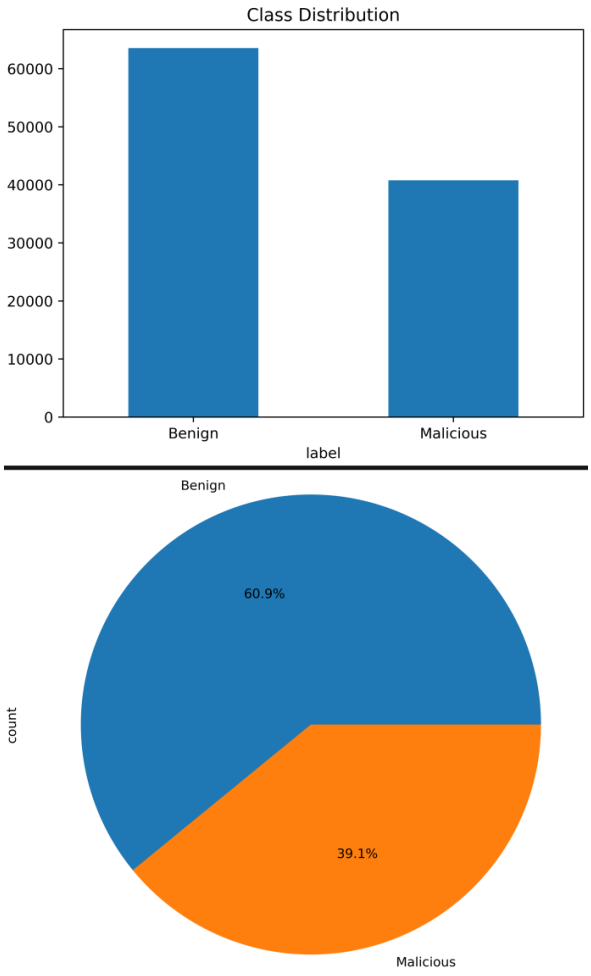

Fig. 2. Classification of attack in the dataset

Digital Commons Data is a comprehensive, cloud-based module that is fully supported and provides all essential functionalities required for establishing an institutional research data management program without necessitating further technical investment. All aspects of software maintenance, configuration, and implementation are overseen by Elsevier teams, which conserves valuable time for users and diminishes the reliance on local IT support. This dataset is produced using the Mininet emulator and is tailored for traffic classification utilizing machine learning and deep learning methodologies. Some of the important columns are:

- 'dt' (timestamp);
- 'src' (source IP);
- 'dst' (destination IP);
- 'pktcount' (number of packets);
- 'bytecount' (number of bytes);
- 'label' (traffic type).

3.4 Preprocessing. This involves transforming raw data into a format that is useful. The categorical class label is transformed into a discrete representation (0,1) through the application of label encoding, with 0 indicating benign traffic and 1 indicating an attack based on DDoS.

3.5 Data Analysis. Fig. 3 depicts the relationship among numerous features within the dataset. The analysis of the correlation matrix reveals several notable relationships between variables. A strong positive correlation exists between 'dt' and 'Pairflow' (72%), suggesting that as 'dt' increases, 'Pairflow' also tends to increase. Similarly, there is a notable positive correlation between 'pktcount' and 'bytecount' (68%), indicating that higher packet counts are associated with higher byte counts. The very strong correlation between 'pktperflow' and 'byteperflow' (81%) suggests that they measure similar aspects of network traffic. Additionally, 'totkbps' and 'rxkbps' (76%) show a strong correlation, implying that most of the traffic migrates in a predictable manner.

Fig. 3. Correlation between various features in the dataset

3.6 Data Segmentation and Normalization. The dataset was first divided into an 80% training set and a 20% testing set for the purpose of machine learning models. Before training, an essential preprocessing step was carried out to standardize and scale the data using the Standard Scaler from Scikit-Learn**.** This standardization process was crucial to ensure compatibility with the machine learning models and to prevent issues like bias, overfitting, or underfitting

3.7 Classification. The subsequent subsections provide an overview of the classification models that have been utilized.

K-Nearest Neighbor (KNNKNN represents a classification methodology that categorizes test data observations by assessing their closeness to the nearest class neighbors. This method is applied as a semi-supervised learning technique and is utilized to determine the nearest neighbors [22]. It operates on a non-parametric basis to classify samples. The distance between distinct points on the input vector is calculated, and the unlabeled point is subsequently assigned to the neighboring class K. The parameter K is crucial in KNN classification. A larger K results in a prolonged prediction process, which can impact accuracy [23]. KNN is straightforward to comprehend when working with a limited number of predictor variables. For models involving standard data types (such as text), KNN is frequently employed.

Decision Tree (DT): A Decision Tree employs a tree structure, where each leaf node represents a potential solution to a class label based on specific conditions [24]. While decision tree algorithms are mainly utilized for classification tasks, they are also applicable to regression issues. The framework comprises a root node, leaf nodes, and intermediate nodes. At the outset, the algorithm begins at the root node**,** representing the entire dataset**.** During tree construction, an attribute selection measure is used to identify the most suitable attribut**e** within the dataset [25].

Random Forest (RF): This is a supervised learning algorithm which constructs and randomizes a forest made up of several DT. The training process utilizes an ensemble technique known as the bagging method. The bagging method combines multiple learning models to enhance overall accuracy and provide better results**.** In this context, RF generates numerous decision trees (DTs) and combines them to achieve precise predictions. Random Forests can be applied to both classification and regression tasks [26]. The algorithm extracts bootstrap samples from the provided dataset. For each of these samples, an unpruned classification [27] or regression tree is developed. Rather than choosing the optimal split from all predictors, randomly selected samples are utilized to ascertain the best split. The subsequent phase involves predicting new data by aggregating the predictions from various trees, leading to an approximate error prediction. Important factors to consider include refraining from making predictions based on bootstrap samples and calculating the error rate following model evaluation.

Results of the development of machine-based intrusion detection

A variety of ML algorithms were utilized, including K-Nearest Neighbours (KNN), Random Forest (RF), and Decision Tree (DT) algorithms. Various classification metrics were calculated to guarantee the optimal functioning of these models, such as accuracy, precision, recall, F1-score, and ROC curve/values. Moreover, an analysis of feature importance using SHAP (SHapley Additive explanations) was conducted to thoroughly investigate how each feature impacts the decision-making process of the model in making predictions. This extensive assessment allowed us to pinpoint the most effective model and the significant features affecting its performance, thus improving the accuracy and dependability of our intrusion detection system.

4.1 K-Nearest Neighbors (KNN) Model Analysis. The SHAP feature importance analysis for the KNN model (Fig. 4) reveals that network throughput metrics overwhelmingly dominate the predictions of the model, with ***tx_kbps*** and ***rx_kbps*** each contributing the highest average impact of **+0.19**, followed closely by ***tot_kbps*** at **+0.11**, indicating that overall traffic volume and directional bandwidth are the primary drivers of the model's decisions.

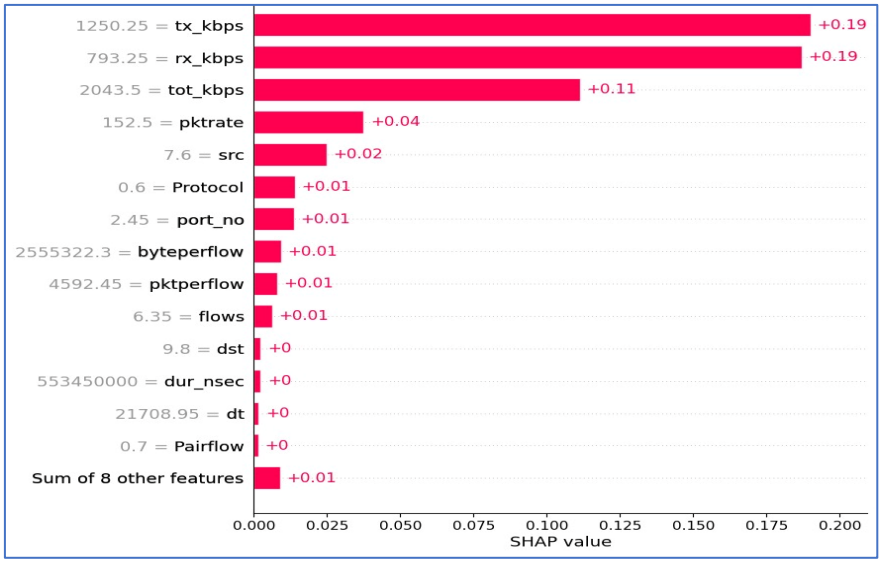

Fig. 4. SHAP bar plot for KNN model feature importance

Packet rate (***pktrate***) shows a moderate influence (+0.04), while features such as source address (*src*), protocol, port number, byteperflow, pktperflow, flows, and destination (***dst***) provide only minimal contributions (ranging from **+0.01 to +0.02**), and several others—including flow duration (***dur_nsec***), ***dt***, and ***Pairflow*** exhibit essentially negligible effects (+**0**). Collectively, the remaining eight low-impact features add just **+0.01**, underscoring that the performance of the KNN model relies heavily on a small subset of bandwidth-related features and suggesting that focusing feature selection on ***tx_kbps***, ***rx_kbps***, ***tot_kbps***, and ***pktrate*** could substantially simplify the model, reduce complexity, and potentially improve efficiency without significant loss in predictive accuracy.

The K-Nearest Neighbors (KNN) model demonstrates high performance, achieving (Table 1):

- Accuracy**:** 98.21%.
- Precision**:** 98.16%.
- Recall**:** 98.10%.
- F1-score**:** 98.1%.

The ROC curve, exhibiting an AUC of 1.000, signifies outstanding discriminative capability, demonstrating the model's proficiency in differentiating between classes. The confusion matrix illustrates the model's performance, which includes (Table 2):

- 8,028 true positives.
- 12,469 true negatives.
- 170 false positives.
- 202 false negatives.

Additional metrics include:

- True Positive Rate (TPR): 0.9754.
- False Positive Rate (FPR): 0.0134.
- False Negative Rate (FNR): 0.0245.
- True Negative Rate (TNR): 0.9865.

These findings further underscore the model's strength in accurately classifying benign and malicious traffic, as depicted in the accompanying ROC curve and confusion matrix visualizations (Fig. 5, 6).

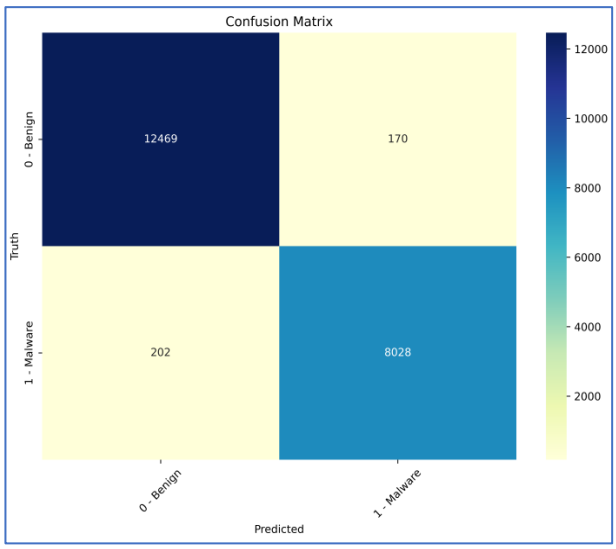

Fig. 5. K-Nearest Neighbors confusion matrix

Table1 – **Experimental results for KNN**

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.9821	0.9816	0.9810	0.9813

Table2 – **Confusion matrix for KNN**

Model	TPR	FPR	FNR	TNR
KNN	0.9754	0.0134	0.0245	0.9865

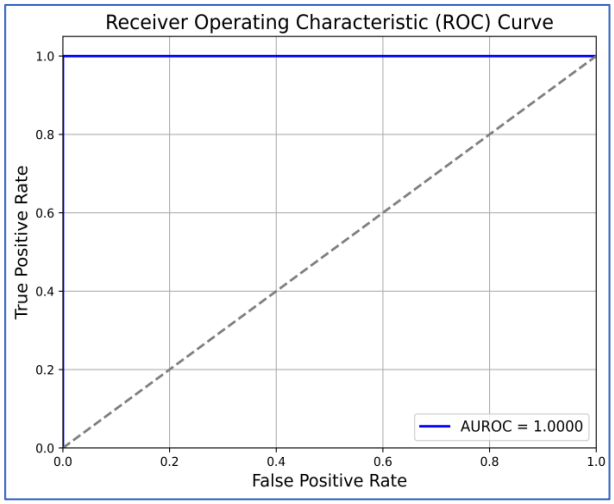

Fig. 6. K-Nearest Neighbour ROC Curve

4.2 Decision Tree (DT) Analysis. The SHAP feature importance results for the decision tree model (Fig. 7) shows that its predictions are dominated by a small set of packet- and flow-level features. In particular, ***packetins*** (mean SHAP = +0.24) and ***byteperflow*** (+ 0.19) are the most influential drivers of the model's decisions, with ***protocol*** contributing moderately (+0.07). All other features exhibit negligible importance, indicating minimal impact on prediction. Overall, the model relies heavily on a few key aggregation features while largely ignoring traffic volume, duration, and address-related variables, suggesting that focused feature selection around these dominant features could improve interpretability, reduce complexity, and maintain predictive performance.

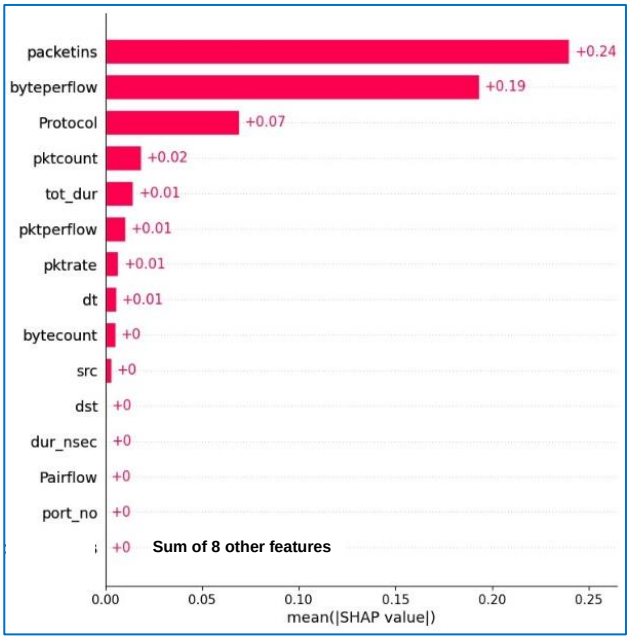

Fig. 7. SHAP bar plot
for Decision Tree model feature importance

The Decision Tree Classifier model demonstrates flawless performance metrics, achieving (Table 3):

- Recall**:** 1.0.
- F1-score**:** 1.0.
- ROCAUC**:** 1.0.
- Accuracy**:** 1.0.
- Precision**:** 1.0.

The confusion matrix corroborates this with (Table 4):

- 8,230 true positives.
- 12,639 true negatives.
- 0 false positives.
- 0 false negatives.

Additional metrics include:

- False Positive Rate (FPR): 0.
- False Negative Rate (FNR): 0.
- True Positive Rate (TPR): 1.0.
- True Negative Rate (TNR): 1.0.

These metrics highlight the model's exceptional performance on the test set, withno indication of data leakage (Fig. 8, 9). However, the perfect scores suggest a need to ensure the test set is representative to avoid potential overfitting concerns.

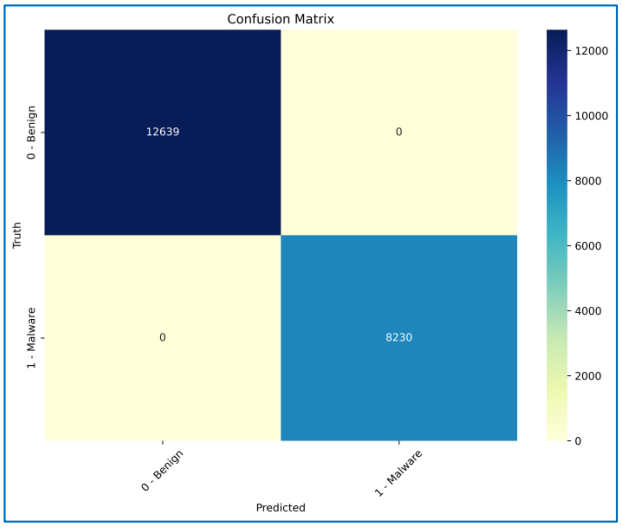

Fig. 8. Decision Tree confusion matrix

Table 3 – **Experimental results for DT**

Model	Accuracy	Precision	Recall	F1-Score	ROC
DT	1.0000	1.0000	1.0000	1.0000	1.0000

Table 4 – **Confusion matrix for DT**

Model	TPR	FPR	FNR	TNR
DT	0.9998	0.0001	0.0001	0.9998

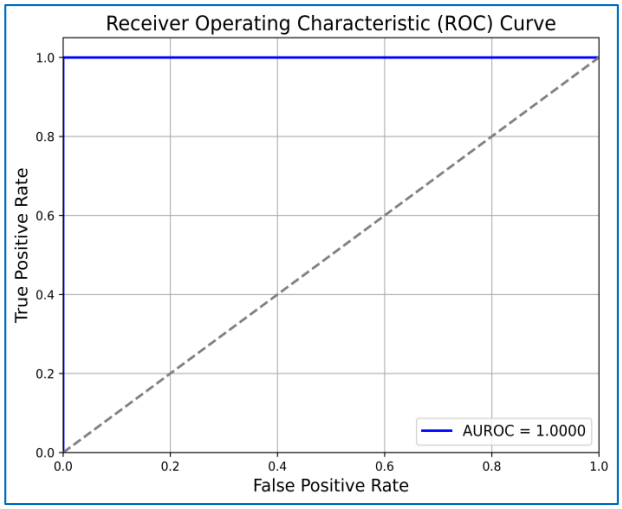

Fig. 9. Decision Tree ROC Curve

4.3 Random Forest (RF) Analysis. The SHAP bar plot (Fig. 10), shows that ***bytecount*** is the most influential feature (mean SHAP ≈ **+0.11**), followed by ***byteperflow*** (**+0.08**) and ***pktcount*** (**+0.07**). ***pktperflow***, ***tot_dur***, and ***packetins*** contribute moderately, each with mean SHAP values around **+0.05**. Features such as ***pktrate*** and ***protocol*** have lower influence (≈ **+0.03**), while ***pairflow*** and ***dt*** show minimal impact (≈ **+0.02**). Address- and duration-related features (***src***, ***flows***, ***dur***, ***dur_nsec***) and the remaining features collectively contribute negligibly (≈ **+0.01**). Overall, the RF model relies mainly on traffic volume and flow-level features, with many features having little to no effect on predictions.

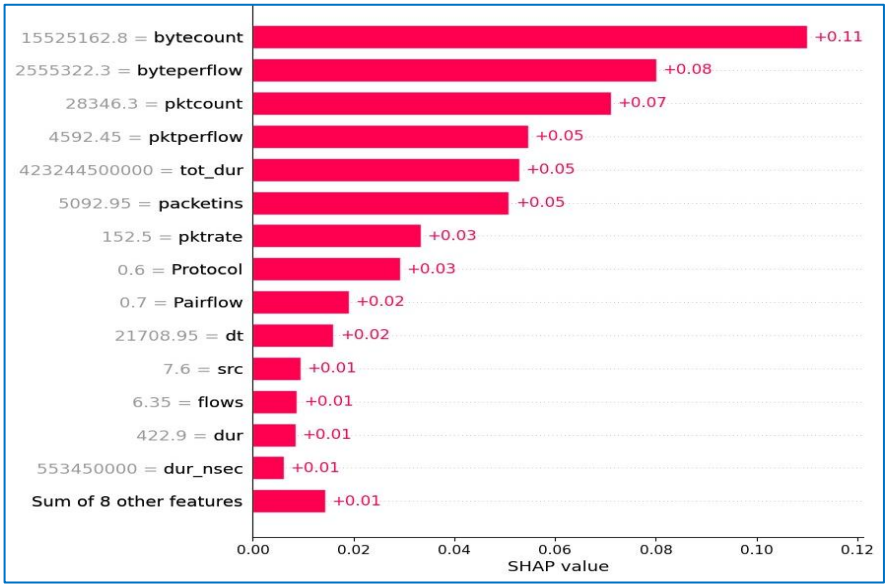

Fig. 10. SHAP bar plot for Random Forest model feature importance

Overall, while ‘packetins’ stands out as the key feature**,** a significant number of features have little to no impact on the model.

The Random Forest Classifier model exhibits perfect performance metrics, achieving (Table 5):

- Accuracy**:** 1.0000.
- Precision**:** 1.0000.
- Recall**:** 1.0000.
- F1-score**:** 1.0000.
- ROCAUC**:** 1.0000.

The confusion matrix confirms this (Table 6), with:

- 8,230 true positives.
- 12,639 true negatives.
- 0 false positives.
- 0 false negatives.

Additional metrics include:

- True Positive Rate (TPR):[0, 1].
- True Negative Rate (TNR):[0, 1].
- False Positive Rate (FPR):[0, 1].
- False Negative Rate (FNR)**:** [1, 0].

These metrics suggest the model might be memorizing the training data rather than generalizing well, as reflected in the attached ROC curve and confusion matrix charts (Fig. 11, 12).

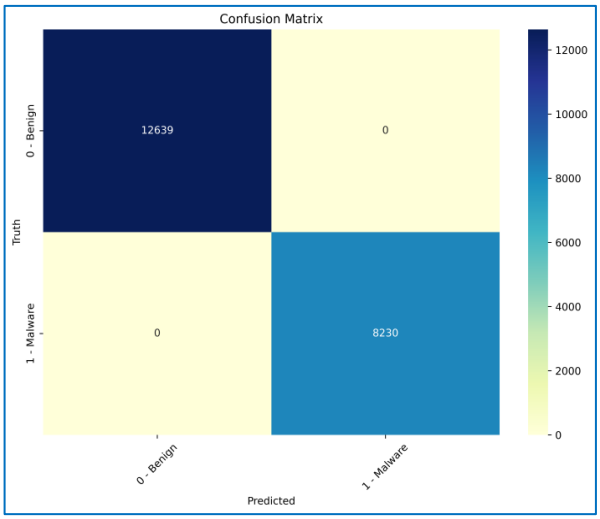

Fig. 11. Random Forest confusion matrix

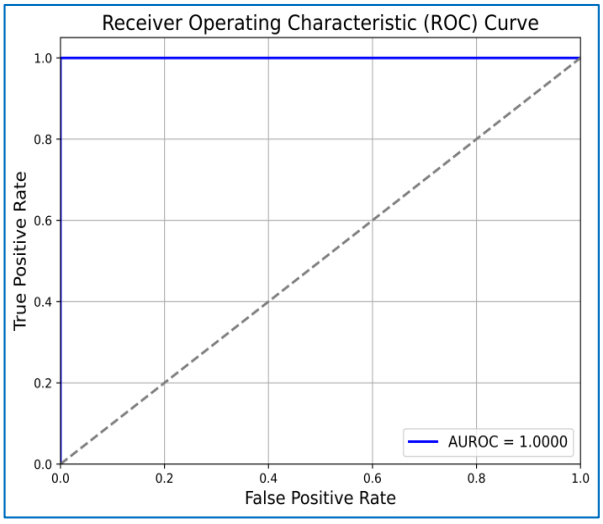

Fig. 12. Random Forest ROC Curve

Table 5 – **Experimental results for RF**

Model	Accuracy	Precision	Recall	F1-Score	ROC
RF	1.0000	1.0000	1.0000	1.0000	1.0000

Table 6 – **Confusion matrix for RF**

Model	TPR	FPR	FNR	TNR
RF	0.9998	0.0001	0.0001	0.9998

4.4 Comparative Analysis: In Table 7 (Final – Final Prediction (Ensemble Majority Voting)), the KNN model demonstrates strong performance, achieving:

- Accuracy: 0.9821.
- Precision: 0.9816.
- Recall: 0.9810.
- F1-score: 0.9813.
- ROC AUC: 1.0000.

These metrics indicate that KNN effectively identifies both true positives and true negatives. However, the confusion matrix reveals a:

- True Positive Rate (TPR): 0.9754.
- True Negative Rate (TNR): 0.9865.

Table 7 – **Performance comparison of the different models used**

Model	Acc	Precision	Recall	F1-Score	ROC
DT	1.0000	1.0000	1.0000	1.0000	1.0000
RF	1.0000	1.0000	1.0000	1.0000	1.0000
KNN	0.9821	0.9816	0.9810	0.9813	1.0000
Final	1.0000	1.0000	1.0000	1.0000	N/A

While both rates are high, indicating strong performance, the TPR is slightly lower, suggesting a minor underperformance in predicting positive cases compared to negative ones (Fig. 13, Tabl. 8).

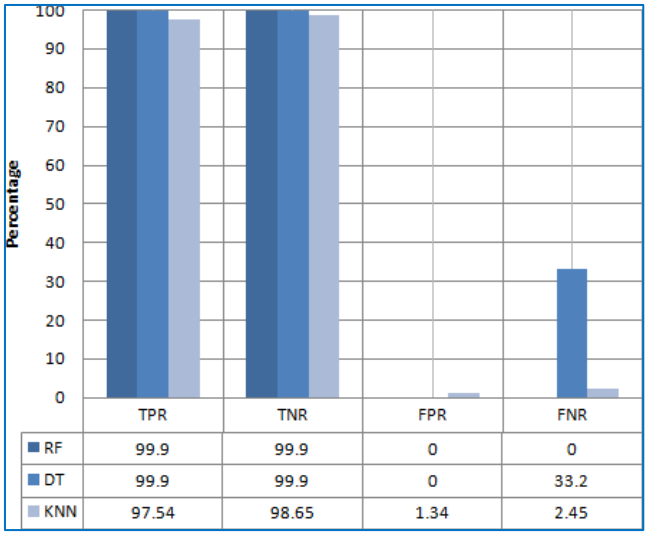

Fig. 13. Accuracy measurement of algorithms used in this work

The Random Forest Classifier achieves seamless scores across all metrics, including:

- Accuracy, Precision, Recall, F1-score, and ROC AUC: 1.0

- Confusion Matrix TPR: 0.9998.
- Confusion Matrix TNR: 0.9998.

Table 8 – **Accuracy measurement of algorithms used in this work**

Model	TPR	FPR	FNR	TNR
KNN	0.9754	0.0134	0.0245	0.9865
DT	0.9998	0.0001	0.0001	0.9998
RF	0.9998	0.0001	0.0001	0.9998

The Random Forest Classifier achieves seamless scores across all metrics, including:

- Accuracy, Precision, Recall, F1-score, and ROC AUC: 1.0
- Confusion Matrix TPR: 0.9998.
- Confusion Matrix TNR: 0.9998.

This indicates the test set flawless classification demonstrating the model's ability to generalize well and perfectly predict both classes. Similarly, the Decision Tree Classifier achieves perfect scores:

- Accuracy, Precision, Recall, F1-score, and ROC AUC: 1.0000.
- TPR and TNR values identical to the Random Forest Classifier.

This reflects the Decision Tree's ability to handle the classification task without errors on the test set. In summary (Fig. 14):

- All models exhibit excellent performance, with Random Forest and Decision Tree classifiers achieving perfect predictions on the test set.
- The KNN model also performs exceptionally well, though it has a slight imbalance in class predictions.

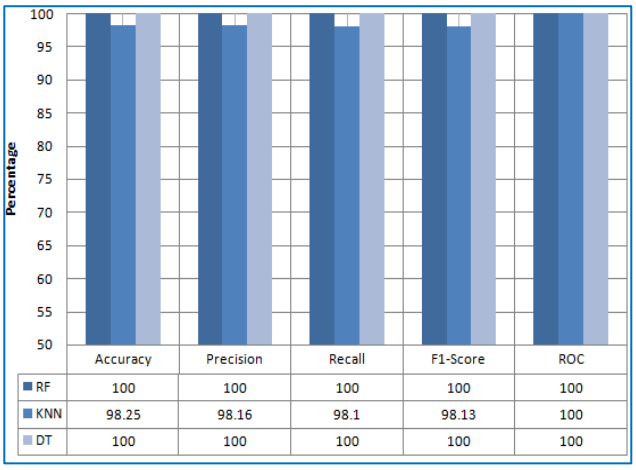

Fig. 14. Comparison of different machine learning algorithms used

- All models demonstrate strong generalization to the test data.

4.5 Comparison. This section presents a comparative analysis of the proposed method alongside other recent advancements in machine learning for detecting attacks in cloud networks, as illustrated in Table 9. The accuracy evaluation metric serves as the basis for comparison (Fig. 15).

Conclusions

1. This study seeks to forecast the likelihood of a Distributed Denial of Service (DDoS) attack within a Software-Defined Network (SDN) operating in a cloud computing context. The methodology proposed is depicted in Fig. 1.

Table 9 – **Work comparison of the proposed model against other close rivals**

Research Work	Dataset	Model	Average Accuracy Score (%)
[16]	Self-generated	SVC and RF	98.8
[17]	InSDN	CNN	96.43
[19]	Self-generated	RF, DT, ND, K-NN, and SVM	99.88
Proposed Model	SDN-DDoS	KNN, DT, and RF	100

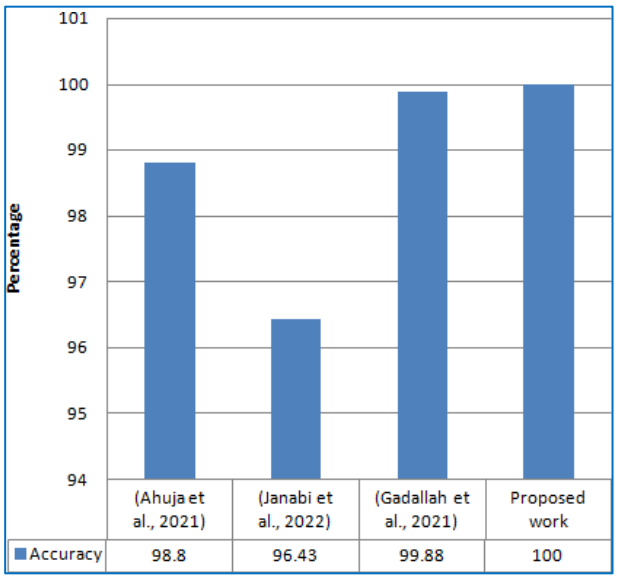

Fig. 15. Detection rate comparison of different methods

2. Our ensemble machine learning framework, which includes RF, KNN, and DT, attained a perfect accuracy rate of 100% on the test dataset. Both the DT and RF algorithms exhibited comparable performance, each achieving 100% accuracy and significantly surpassing the KNN algorithm.

3. The performance results of Random Forest align with the findings of [28], reinforcing the suitability of Random Forest for the classification of DDoS attack events in a system. The K-Nearest Neighbor algorithm also demonstrated commendable performance, reaching an accuracy of 98.21% in detecting DDoS attacks.

4. Nonetheless, as shown in Table 9, the proposed ensemble machine learning model for DDoS attack detection has been evaluated against leading models in the literature and has been ranked as the most accurate.

5. The findings from this research demonstrate that machine learning (ML) models that applies ensemble techniques can potentially enhance the accuracy of

intrusion detection in cloud environments. We can find the application of the findings in cloud service providers (CSPs), enterprises, and security teams that integrate ML-driven detection modules.

6. The tech ecosystem stands to benefit from those findings as when it is applied in cloud environment such as AWS SageMaker, the respective enterprise experience reduced financial losses from cyberattacks, greater trust in cloud services and more secure digital infrastructure supporting internet services and products.

7. However, there is a concern about the privacy of dataset information. Future research could explore intrusion detection and the use of reinforcement learning for autonomous threat response.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Mhamdi, L. and Isa, M.M. (2024), "Securing SDN: Hybrid auto encoder random forest for intrusion detection and attack mitigation", *Journal of Network and Computer Applications.*, vol. 225, article number 103868, doi: https://doi.org/10.1016/j.jnca.2024.103868
2. Rajadurai, H. and Gandhi, U.D. (2022), "A stacked ensemble learning model for intrusion detection in wireless network", *Neural computing and applications*, vol. 34, pp. 15387–15395, doi: https://doi.org/10.1007/s00521-020-04986-5
3. Amaran, S. and Mohan, R.M. (2021), "Intrusion detection system using optimal support vector machine for wireless sensor networks", *2021 International Conference on Artificial Intelligence and Smart Systems* (ICAIS), IEEE, pp. 1100–1104, doi: https://doi.org/10.1109/ICAIS50930.2021.9395919
4. Sharma, S., Zavarsky, P. and Butakov, S. (2020), "Machine learning based intrusion detection system for web-based attacks", *2020 IEEE 6th Intl Conference on Big Data Security on Cloud* (Big Data Security), IEEE, pp. 227–230, doi: https://doi.org/10.1109/BigDataSecurity-HPSC-IDS49724.2020.00048
5. Elmrabit, N., Zhou, F., Li, F. and Zhou, H. (2020), "Evaluation of machine learning algorithms for anomaly detection", *2020 international conference on cyber security and protection of digital services* (cybersecurity), IEEE. pp. 1–8, doi: https://doi.org/10.1109/CyberSecurity49315.2020.9138871
6. Gao, X., Shan, C., Hu, C., Niu, Z. and Liu, Z. (2019), "An adaptive ensemble machine learning model for intrusion detection", Ieee Access 7, pp. 82512– 82521, doi: https://doi.org/10.1109/ACCESS.2019.2923640
7. Venkatesan, S. (2023), "Design an intrusion detection system based on feature selection using ml algorithms", *Mathematical Statistician and Engineering Applications*, vol. 72(1), pp. 702–710, available at: https://www.philstat.org/index.php/MSEA/article/view/2000
8. Alduailij, M., Khan, Q.W., Tahir, M., Sardaraz, M., Alduailij, M. and Malik, F. (2022), "Machine-learning-based DDoS attack detection using mutual information and random forest feature importance method, *Symmetry*, vol. 14, 1095, doi: https://doi.org/10.3390/sym14061095
9. Jaber, A.N. and Rehman, S.U. (2020), "FCM–SVM based intrusion detection system for cloud computing environment", *Cluster Computing*, vol. 23, pp. 3221– 3231, doi: https://doi.org/10.1007/s10586-020-03082-6
10. Aldallal, A. and Alisa, F. (2021), "Effective intrusion detection system to secure data in cloud using machine learning", *Symmetry*, vol. 13, 2306, doi: https://doi.org/10.3390/sym13122306
11. Singh, P. and Ranga, V. (2021), "Attack and intrusion detection in cloud computing using an ensemble learning approach", *International Journal of Information Technology*, vol. 13, pp. 565–571, doi: https://doi.org/10.1007/s41870-020-00583-w
12. Wei, J., Long, C., Li, J. and Zhao, J. (2020), "An intrusion detection algorithm based on bag representation with ensemble support vector machine in cloud computing", *Concurrency and Computation: Practice and Experience*, vol. 32, e5922, doi: https://doi.org/10.1002/cpe.5922
13. Mohmand, M.I., Hussain, H., Khan, A.A., Ullah, U., Zakarya, M., Ahmed, A., Raza, M., Rahman, I.U. and Haleem, M. (2022), "A machine learning based classification and prediction technique for DDoS attacks, *IEEE Access*, vol. 10, pp. 21443–21454, doi: https://doi.org/10.1109/ACCESS.2022.3152577
14. Firdaus, D., Munadi, R. and Purwanto, Y. (2020, "DDoS attack detection in software defined network using ensemble k-means++ and random forest", *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems* (ISRITI), IEEE, pp. 164–169, doi: https://doi.org/10.1109/ISRITI51436.2020.9315521
15. Nadeem, M.W., Goh, H.G., Ponnusamy, V. and Aun, Y. (2022), "DDoS detection in SDN using machine learning techniques*", Computers, Materials & Continua*, vol. 71(1), pp. 771–789, doi: https://doi.org/10.32604/cmc.2022.021669
16. Ahuja, N., Singal, G., Mukhopadhyay, D. and Kumar, N. (2021), "Automated DDoS attack detection in software defined networking", *Journal of Network and Computer Applications*, vol. 187, doi: https://doi.org/10.1016/j.jnca.2021.103108
17. Janabi, A.H., Kanakis, T. and Johnson, M. (2022), "Convolutional neural network based algorithm for early warning proactive system security in software defined networks", *IEEE Access*, vol. 10, pp. 14301–14310, doi: https://doi.org/10.1109/ACCESS.2022.3148134
18. Akinwumi, A., Akingbesote, A., Ajayi, O. and Aranuwa, F. (2022), "Detection of distributed denial of service (DDoS) attacks using convolutional neural networks", *Nigerian Journal of Technology*, vol. 41, pp. 1017–1024, doi: https://doi.org/10.4314/njt.v41i6.12
19. Gadallah, W.G., Omar, N.M. and Ibrahim, H.M. (2021), "Machine learning - based distributed denial of service attacks detection technique using new features in software-defined networks", *International Journal of Computer Network & Information Security*, vol. 13, pp. 15–27, doi: https://doi.org/10.5815/ijcnis.2021.03.02
20. Babaei, A., Kebria, P.M., Dalvand, M.M. and Nahavandi, S.(2023), "A review of machine learning-based security in cloud computing", *arXiv preprint*, arXiv:2309.04911, doi: https://doi.org/10.48550/arXiv.2309.04911

21. Housman, O.G., Isnaini, H. and Sumadi, F.D.S. (2020), “SDN-DDoS (ICMP,TCP,UDP)”, *Mendeley Data*, Version 1, doi: https://doi.org/10.17632/hkjbp67rsc.1
22. Peterson, L.E. (2009), “K-nearest neighbor”, *Scholarpedia*, vol. 4(2), doi: http://dx.doi.org/10.4249/scholarpedia.1883
23. Batista, G. and Silva, D.F (2009), “How K-nearest neighbor parameters affect its performance”, *Semantic Scholar*, Corpus ID: 16606615, pp. 1–12, available at: https://api.semanticscholar.org/CorpusID:16606615
24. Okandeji, A., Odeyinka, O., Sogbesan, A. and Ogunye, N. (2022), “A comparative analysis of haemoglobin variants using machine learning algorithms”, *Nigerian Journal of Technology*, vol. 41, pp. 789–796, doi: https://doi.org/10.4314/njt.v41i4.16
25. Vemulapalli, S., Sushma Sri, M., Varshitha, P., Kumar, P., Vinay, T. (2024), “An experimental analysis of machine learning techniques crop for recom- mendation”, *Nigerian Journal of Technology*, vol. 43(2), pp. 301–308, doi: https://doi.org/10.4314/njt.v43i2.13
26. Nkiama, H., Said, S.Z.M. and Saidu, M. (2016), “A subset feature elimination mechanism for intrusion detection system”, *International Journal of Advanced Computer Science and Applications*, vol. 7, doi: https://dx.doi.org/10.14569/IJACSA.2016.070419
27. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H. and Wang, C. (2018), “Machine learning and deep learning methods for cybersecurity”, *IEEE access*, vol. 6, pp. 35365–35381, doi: https://doi.org/10.1109/ACCESS.2018.2836950
28. Shen, Z., Zhang, Y. and Chen, W. (2019), “A bayesian classification intrusion detection method based on the fusion of PCA and LDA”, *Security and Communication Networks*, vol. 2019, pp. 1–11, doi: https://doi.org/10.1155/2019/6346708

Received (Надійшла) 31.08.2025
Accepted for publication (Прийнята до друку) 07.01.2026

ВІДОМОСТІ ПРО АВТОРІВ / ABOUT THE AUTHORS

Ісонг Акваено – аспірант кафедри комп’ютерної інженерії університету Уйо, Уйо, Нігерія;
Akwaeno Isong – PhD student of Computer Engineering Department, University of Uyo, Uyo, Nigeria;
e-mail: akwaenoi@gmail.com; ORCID Author ID: https://orcid.org/0009-0001-7613-8343.

Стівен Бліс Утібе-Абасі – доктор філософії, доцент, доцент кафедри комп’ютерної інженерії Університету Уйо, Уйо, Нігерія;
Bliss Utibe-Abasi Stephen – PhD, Associate Professor, Associate Professor of Computer Engineering Department, University of Uyo, Uyo, Nigeria;
e-mail: blissstephen@uniuyo.edu.ng; ORCID Author ID: https://orcid.org/0000-0002-2535-4492;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57202044498.

Асукво Філіп – доктор філософії, професор, професор кафедри комп’ютерної інженерії Університету Уйо, Уйо, Нігерія;
Philip Asuquo – PhD, Professor, Professor of Computer Engineering Department, University of Uyo, Uyo, Nigeria;
e-mail: philipasuquo@uniuyo.edu.ng; ORCID Author ID: https://orcid.org/0000-0003-4888-5461;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=55994284100.

Ігемерезе Чіджіоке Ннанна – головний викладач кафедри комп’ютерної інженерії Федерального політехнічного інституту Некеде, Оверрі, Нігерія;
Chijioke Ihemereze – Principal Lecturer, Principal Lecturer in Computer Engineering Department, Federal Polytechnic Nekede, Owerri, Nigeria;
e-mail: cihemereze@fpno.edu.ng; ORCID Author ID: https://orcid.org/0009-0004-0323-1303.

Енанг Імо Окон – викладач ІІ категорії кафедри комп’ютерної інженерії Федерального політехнічного інституту Некеде, Оверрі, Нігерія;
Imoh Enang – Lecturer II, Lecturer II in Computer Engineering Department, Federal Polytechnic Nekede, Owerri, Nigeria;
e-mail: iudoh@fpno.edu.ng; ORCID Author ID: https://orcid.org/0009-0008-0883-8744.

Машинне навчання для виявлення вторгнень у хмарних обчисленнях

А. Ісонг, Б. У.-А. Стівен, Ф. Асукво, Ч. Н. Ігемерезе, І. О. Енанг

Анотація. В умовах сучасного технологічно поєднаного світу у хмарних обчислювальних середовищах використовується передова мережева технологія, відома як програмно-конфігуровані мережі (SDN), щоб підвищити ефективність управління мережею. Однак централізована природа SDN робить її вразливою до DDoS-атак. У цьому дослідженні представлено метод для виявлення DDoS-атак у середовищі хмарних обчислень. Дослідження спрямоване на застосування ансамблевого підходу машинного навчання для статистичного розпізнавання DDoS-атак у хмарному мережевому трафіку, класифікуючи їх як шкідливі або нешкідливі. Різні алгоритми машинного навчання, включаючи K-ближчих сусідів, випадковий ліс (RF) та дерево рішень (DT), були використані як базові класифікатори в запропонованій ансамблевій моделі машинного навчання. Для оцінки ефективності базових класифікаторів було використано набір даних SDN–DDoS-атак. Класифікатори були навчені на 80% даних і протестовані на 20%. Результати експерименту показали, що класифікатори RF та DT досягли точності 100%, тоді як класифікатор K-ближчих сусідів забезпечив точність 98,21%. Ансамблева модель машинного навчання застосувала метод більшості голосів для фінального прогнозу та досягла точності 100% на тестовому наборі, ставши найкращою порівняно з еталонними моделями.

Ключові слова: хмарні обчислення; класифікація атак; машинне навчання; виявлення загроз; IaaS; PaaS; SaaS; система виявлення вторгнень; штучний інтелект; глибоке навчання; відбір ознак; алгоритми класифікації; виявлення аномалій.