Intelligent Systems with Applications 24 (2024) 200442

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

Enhanced intrusion detection model based on principal component analysis and variable ensemble machine learning algorithm

Ayuba John [a,*], Ismail Fauzi Bin Isnin [b], Syed Hamid Hussain Madni [c], Farkhana Binti Muchtar [b]

[a] Faculty of Computing, Federal University Dutse, Jigawa State, Nigeria
[b] Faculty of Computing, Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia
[c] School of Electronic & Comp. Sc, University of Southampton, Johor Bahru, Malaysia

ARTICLE INFO

Keywords:
Network security
Intrusion detection system
Classification
Detection
and Machine Learning Algorithm

ABSTRACT

The intrusion detection system (IDS) model, which can identify the presence of intruders in the network and take some predefined action for safe data transit across the network, is advantageous in achieving security in both simple and advanced network systems. Several IDS models have various security problems, such as low detection accuracy and high false alarms, which can be caused by the network traffic dataset's excessive dimensionality and class imbalance in the creation of IDS models. Principal Component Analysis (PCA) has proven to be a helpful feature selection technique for dimensionality reduction. As a result, because it is a linear transformation, it has challenges capturing non-linear relationships between feature properties in the network traffic datasets. This paper proposes a variable ensemble machine learning method to solve the problem and achieve a low variance model with high accuracy and low false alarm. First, PCA is combined with the AdaBoost ensemble machine learning algorithm, which acts as stagewise additive modelling to compensate for PCA's deficiency in feature selection in network traffic by minimizing the exponential loss function. Secondly, PCA is used for feature selection, and a LogitBoost classifier algorithm can be used for multiclass classification and acts as an additive tree regression to compensate for the PCA's weakness by minimizing the Logistic Loss to provide an optimal classifier output. Finally, the low variance ability of RandomForest, which employs the bagging approach, is applied to eliminate overfittings. The experiments of the IDS model developed from the proposed methods were evaluated on the WSN-DS, NSL-KDD, and UNSW-N15 datasets. The performance of the methods, PCA with AdaBoost, on the WSN-DS dataset has an accuracy score of 92.3 %, an 89.0 % accuracy score on the NSL-KDD dataset, and a 67.9 % accuracy score on UNSW-N15, which is the least accurate score. PCA and RandomForest surpassed them by scoring 100 % accuracy on all three datasets. PCA and Bagging have an accuracy score of 99.8 % on the WSN-DS dataset, 100 % on the NSL-KDD dataset, and 93.4 % on the UNSW-N15 dataset. In comparison, PCA and LogitBoost have an accuracy score of 98.9 % on the WSN-DS dataset, 100 % on the NSL-KDD dataset, and 88.7 % on the UNSW-N15 dataset.

1. Introduction

Numerous researchers have leveraged artificial intelligence to develop Intrusion Detection System (IDS) models, significantly enhancing their performance in defending network systems against cyber threats (Awotunde & Misra, 2022; Guarascio et al., 2022; Muneer et al., 2024). An IDS analyses network traffic raises alarms when intrusions are detected and monitors for ongoing intrusions within the network system (Ashiku & Dagli, 2021; Gassais et al., 2020). System administrators use IDS to identify threats, ensuring a secure environment for users' accounts, network facilities, personal records, and passwords (Gajewski et al., 2019; Kizza, 2024). Feature selection techniques and classification algorithms are crucial steps in IDS development. Various machine learning algorithms, including conventional, ensemble, and deep learning methods, have been employed for classification (Chen et al., 2019; Mohammed & Kora, 2023; Wang et al., 2021). Feature selection is routinely used in the preprocessing stage to improve classifier performance (Remeseiro & Bolon-Canedo, 2019).

Various issues have been identified in existing research on intrusion detection using machine learning algorithms (Al-Janabi et al., 2021;

* Corresponding author.
E-mail addresses: john@graduate.utm.my, ayuba.john@fud.edu.ng (A. John), ismailfauzi@utm.my (I.F.B. Isnin), s.h.h.madni@soton.sc.uk (S.H.H. Madni), farkhana@utm.my (F.B. Muchtar).

https://doi.org/10.1016/j.iswa.2024.200442
Received 9 February 2024; Received in revised form 28 August 2024; Accepted 14 September 2024
Available online 21 September 2024

Kocher & Kumar 2021; Mishra et al., 2018; Saranya et al., 2020). These include noise, high dimensionality, class imbalance, bias in detecting new threats, poor feature selection, limited storage, deviations in learning patterns, and high computational complexity (Bao et al., 2020; Thudumu et al., 2020; Yang et al., 2022). To address these shortcomings, some researchers have sought to improve detection accuracy by transitioning from conventional machine learning methods like support vector machines (SVM) and decision trees (DT) to ensemble techniques such as random forests (RF) and Bagging (Mafarja et al., 2023; Sothe et al., 2020). Ensemble learning combines multiple models to create more accurate and stable predictions than individual models can achieve (Ganaie et al., 2022; Nti et al., 2020). Many researchers have employed principal component analysis (PCA) to tackle the dimensionality problem in network traffic datasets during feature selection, recognizing its effectiveness over earlier techniques, especially in anomaly detection (Di Mauro et al., 2021; Selvakumar & Muneeswaran, 2019). However, PCA is a linear transformation which struggles to capture non-linear correlations between feature attributes (Li et al., 2020; Lucchese et al., 2020; Uddin et al., 2021). Since most features in network traffic datasets are non-linearly correlated, using PCA for feature selection in an intrusion detection system may result in a high number of false alarms and low accuracy (Al-Fawa'reh et al., 2022; Di Mauro et al., 2021; Zhang et al., 2022b).

The significant contributions of this research paper include developing an IDS model that minimizes the Logistic loss function of LogitBoost, leveraging the low variance capability of Random-Forest's bagging approach to eliminate overfitting, and utilizing the exponential loss function of AdaBoost to address PCA's inability to capture non-linear relationships among network traffic feature attributes. The model detects intrusions in three benchmark network traffic datasets, WSN-DS, NSL-KDD, and UNSW-N15, using ensemble machine learning algorithms such as LogitBoost, AdaBoost, Bagging, and Random-Forest integrated in variable ensemble selection.

The following sections of this work are organized. Section 2 discussed related works. Section 3 describes the proposed technique. Section 4 presented the results analysis and discussion; Section 5 provided the conclusion; Section 6 provided the acknowledgement; Section 7 provided the data availability statement; and Section 8 declared any conflicts of interest.

2. Related works

Singh and Vigila (2023) proposed a Principal Component Analysis (PCA) and Fuzzy extreme learning machine classifier algorithm to develop an Intrusion Detection System(IDS) model to deal with high execution time and low detection accuracy issues in the model, which achieved high detection accuracy but produced high computational overhead. Hossain and Islam (2023b) developed a correlation analysis using mutual information principal component analysis and a multiple ensemble classifier for a model that can detect novelty attacks. It can detect several attacks, but it takes a long time to train and increases the computing complexity of the model. Udas et al. (2022) created an intrusion detection system model with combined algorithms as classifiers (recurrent neural network, bidirectional long-short-term memory gated recurrent unit) and a PCA for dimensionality reduction, significantly improving detection accuracy and reducing model complexity. Still, overfitting occurred due to the inability to derive knowledge from the non-linearity of the data. Ravi et al. (2022) proposed a model built using kernel PCA and a recurrent neural network that was able to identify an optimal feature, improving the model's accuracy but resulting in significant computational complexity. Lv et al. (2020) offer a hybrid kernel extreme learning machine and kernel principal component analysis to construct a model with better accuracy and reduced computing time. However, it fails to recognize some attack classes and overfits the model. Majidian et al. (2023) offered a PCA paired with error correction output codes and an adaptive neuro-fuzzy inference system with a Particle Swarm Optimization algorithm for a model to detect DoS attacks with excellent detection accuracy but at the expense of increased model training time. Kareem et al. (2023) created an intrusion detection model for application layer DDoS attack detection, which enhanced detection accuracy but had a long computation time.

Ebenezer et al. (2023) proposed using a PCA and a support vector machine with K-Nearest Neighbor classifiers to create an IDS model that has enhanced detection and integrates a Docker prison system to allow it to stop an attack once the IDS model has detected it but cannot identify the attack types. Putra et al. (2023) used a PCA with Truncate singular value decomposition, factor analysis, and fast independent analysis with several conventional machine learning algorithms for an IDS and compared several feature selection techniques, but there is no clear evidence of attack detection. Zhiqiang et al. (2022) presented a framework for detecting an attack node in a wireless sensor network using an upgraded empirical-based component analysis with long short-term memory. Still, they are unable to distinguish attack classes. Al-Fawa'reh et al. (2022) developed a PCA with a deep neural network algorithm to address the problem of long-term attack detection and the inability to identify zero-day attacks. It has improved the model's performance detecting DDoS and DoS attacks, but the training time is much longer. Guezzaz et al. (2022) created an IDS model utilizing PCA and a K-Nearest Neighbor classifier to improve attack detection while increasing the model's training time. Rajadurai and Gandhi (2021) employ a PCA with a deep learning algorithm to create an IDS model that improves attack detection and classifies attacks but cannot detect unknown attacks. Camacho et al. (2019) used a PCA with a group-wise technique to develop an intrusion model with improved feature selection and attack detection; nevertheless, the model lacks expert knowledge to tune depending on security experiences. Salman et al. (2018) proposed combining a PCA with learning vector quantization and Big data approaches to create an IDS model, which enhanced the PCA's efficiency for feature selection but was not designed to detect attacks. Mishra et al. (2020) presented a PCA with a support vector machine for the IDS model to reduce computing time. Although this improves attack detection and reduces computing time, the training is substantially longer. Osho et al. (2021) created an IDS model utilizing PCA and decision trees; it increased the attack detection rate, but the model's efficiency has not been compared to others.

Hossain and Islam (2023a) propose correlation analysis to mutually work with principal component analysis to be used for feature selection and utilize multiple ensemble machine learning algorithms as classifiers to develop an intrusion detection system that can protect the computer system from unauthorized access through obtaining a good result in terms of the evaluation metrics used but did not put into consideration the PCA's inability to capture the non-linearity among the features of the datasets. Singh et al. (2023) introduced a hybrid framework by combining probabilistic principal component analysis for feature selection and using a generalized additive model to create an intrusion detection system that only performs well on wireless sensor network scenarios. Therefore, in this research work, it is of paramount importance to dealt with PCA's issues of inability to capture non-linear features in the datasets and to propose an efficient network security framework for the adequate detection of intrusion attacks.

2.1. Bagging algorithm

Bagging is an ensemble meta-estimator that fits based classifiers on random subsets of the original dataset and aggregates their predictions by voting or averaging to generate a final prediction (Konhäuser et al., 2022). It is a supervised machine learning technique consisting of numerous base models trained separately and in parallel on distinct subsets of training data (Abdoli et al., 2023; González et al., 2020; Hu et al., 2021). Each subgroup is created via bootstrap sampling, randomly selecting data points with replacements (James et al., 2023). The bagged estimator has a lower variance than the original estimate, resulting in a

Algorithm 1
Bagging Classifier.

1. Construct a bootstrap sample $(X_1^*, Y_1^*), \ldots, (X_n^*, Y_n^*)$ by randomly drawing 'n' times with replacements from the data $(X_1, Y_1), \ldots, (X_n, Y_n)$;
2. Compute the bootstrapped estimator $\widehat{g}^*(.)$ By the plug-in principle:
   $\widehat{g}^*(.) = h_n\left((X_1^*, Y_1^*), \ldots, (X_n^*, Y_n^*)\right)(.)$;
3. Repeat steps 1 and 2 'M' times, yielding $\widehat{g}^{*k}(.)(k = 1, \ldots, M)$
   The bagged estimator is $\widehat{g}_{Bag}(.) = M^{-1}\sum_{k=1}^{M}\widehat{g}^{*k}(.)$;

Algorithm 2
RandomForest Classifier.

1. For b = 1 to B:
   Make decision trees using a random sample of the training dataset and develop them into a random forest tree T_b for the bootstrapped data Z^* Of size N by recursively repeating the procedures below for each terminal node of the tree until the minimal node size is reached:
   i. Choose variables at random from the list,
   ii. Choose the best variable/split point from the list.
   iii. Produce an output for each decision tree.
2. Output the ensemble of trees $\{T_b\}_1^B$.
2. Finally, as the final prediction result, choose the most-voted prediction at a new point x and
   let $\widehat{C}_b(x)$ Be the class prediction of the b^{th} Random Forest tree.
   Then, $\widehat{C}_{rf}^B(x) = majority\ vote\ \{\widehat{C}_b(x)\}_1^B$

significant variance reduction if the original estimation is unstable (Barrow et al., 2020; Hillebrand et al., 2021; Kazak & Pohlmeier, 2023). As a result, it is a variance reduction strategy for a base method that performs variable selection and fitting in a linear model on high-dimensional data.

Algorithm 1.

2.2. RandomForest algorithm

The RandomForest algorithm constructs several decision trees on distinct samples, each with a different set of observations, and then selects the majority vote (Valavi et al., 2021). It also employs the bagging approach, which combines parallel modelling and coupled prediction to overcome overfitting (Bakır et al., 2024; Sahoo et al., 2022; Zhang et al., 2021; Zounemat-Kermani et al., 2021). RandomForest bootstrapping is essentially row and feature sampling with a replacement before training the model (Wang et al., 2022). It is slower to compute, but it eliminates the overfitting problem, according to Abdelwahed et al. (2022).

Algorithm 2.

2.3. Boosting algorithm

A boosting algorithm combines a group of weak learners to create a strong learner to reduce training errors by changing the models from high bias to low bias (Zhang et al., 2022c). It works sequentially, with each predictive output model relying on the previous output as an input to the next model, and the final predictive output is regarded as the predictions' output (Asselman et al., 2023; Bentéjac et al., 2021). AdaBoost, Gradient Boosting, and XGBoost are the most frequent boosting instances.

2.4. AdaBoost multiclass classifier

It is a stagewise additive modelling machine learning algorithm that minimizes exponential loss using a multiclass exponential loss function (Alabdulmohsin, 2019; Tanha et al., 2020; Um et al., 2023). Assuming $'n'$ given feature vectors:

$x_1 = (x_{11}, \ldots x_{1p}), \ldots x_n = (x_{n1}, \ldots x_{np})$, where $'p'$ is the size of the feature vectors, and assuming a vector of class labels $y = (y_1, \ldots y_n)$, where;

$y_i \in k = \{-1, 1\}$ is for binary classification and $y_i \in k = \{0, \ldots, c-1\}$, where $'c'$ is the class number of class vector $'t'$ in feature vector $'x_i'$ and given $'h_t'$, which is a weak learner algorithm. Therefore, AdaBoost can be built as follows:

Algorithm 3.

2.5. LogitBoost (Additive logistic regression)

The boosting methodology is used to build a logit model, and the regression method is used as a weak classifier to permit the writing of equations for future prediction in new data (Chu et al., 2020). Thus, it resists overfitting by maximizing an exponential criterion (Jain et al., 2020) equal to the binomial log-likelihood criterion in the second order.

The exponential criterion:

$$J(F) = E\left(e^{-yF(x)}\right) \tag{1}$$

The function $F(x)$ that minimizes $J(F)$ is the symmetric logistic transform of $P(y = 1|x)$, where p is the probability and x is the input variable, $F(x)$ is the additive regression models expressed as follows:

Algorithm 3
AdaBoost Algorithm.

Initializes the weights as $D_1(i) = \frac{1}{n}$ $for\ i = 1, \ldots, n$
For $t = 1, \ldots, T$:
Train the weak learner $h_t(i)$ by the weights D_t,
Choose a confidence value. $\propto_t = \frac{log1 - err_1}{err_t} + \log(c-1)$,
$err_t = \frac{\sum_{i=1} nD_t(i) l(y \neq h_{t(i)})}{\sum_{i=1} nD_{t(i)}}$,
Update $D_{t+1}(i) = \frac{D_t i exp\left(-\propto_t l(y_t \neq h_t(i))\right)}{z_t}$, where z_t It is a normalization factor.
Output the classifier $H(x) = arg_k max \sum_{t=1}^{T} \propto_t l(h_t x = k)$.

Algorithm 4
LogitBoost Algorithm Exponential Criterion.

1. Start with weights $w_i = 1|N,\ i = 1,2,\ldots,\ N,\ F(x) = 0$ and probability estimates $p(x_i) = \frac{1}{2}$;
2. Repeat for $M = 1,2,\ldots,\ M$:
   Compute the working response and weights
   $z_i = \frac{y_i^* - p(x_i)}{p(x_i)(1-p(x_i)}$;
   $w_i = p(x_i)(1 - p(x_i))$;
   Fit the function $f_m(x)$ by a weighted least-squares regression of z_i to x_i using weighted w_i;
3. Update $F(x) \leftarrow F(x) + \frac{1}{2} f_m(x)$ and $p(x) \leftarrow \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}}$;
4. Output the classifier sign $[F(x)] = sign\left[\sum_{M=1}^{M} f_m(x)\right]$;

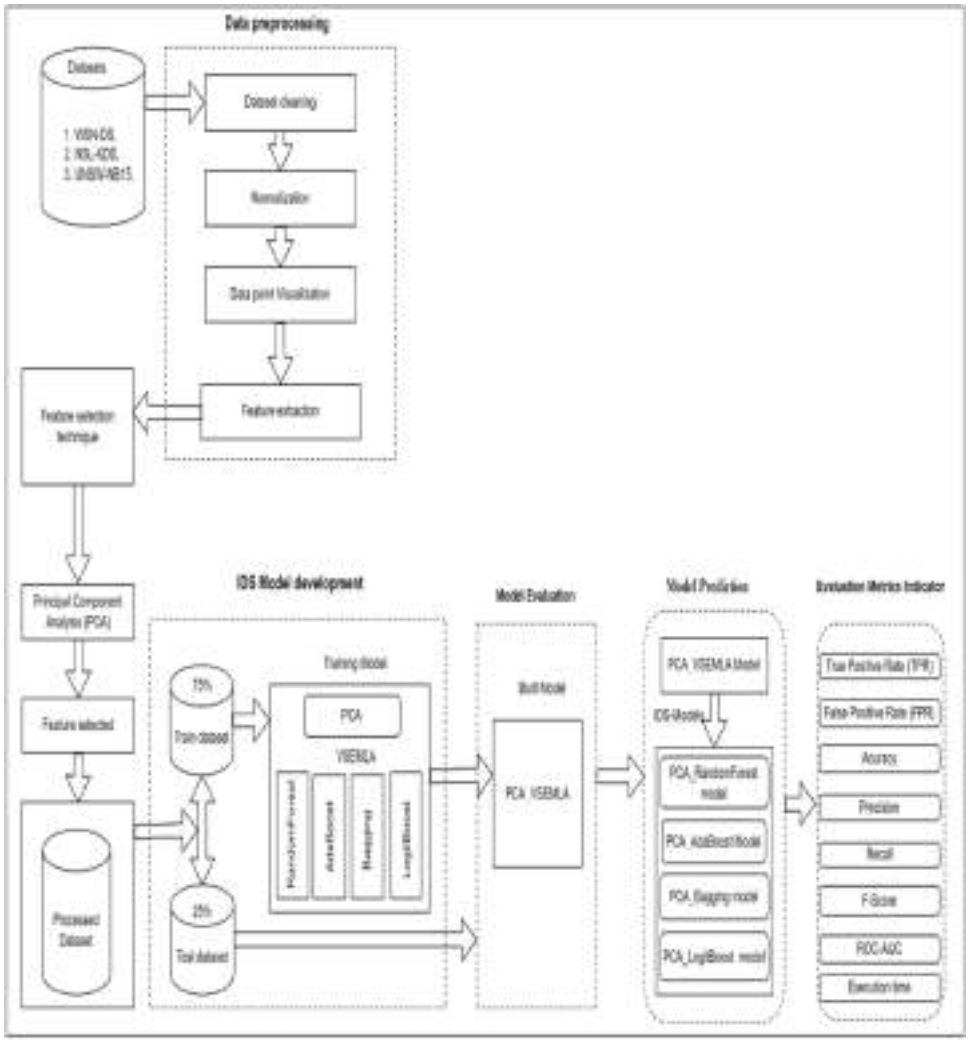

Fig. 1. Architecture of the Proposed IDS Model.

$$F(x) = \sum_{j=1}^{p} f_j(x_j) \tag{2}$$

Where $f_j(x_j)$ is a separate function for each of the probability (p) input variables x_j and from Eq. (1), $E(e^{-yF(x)})$ Is minimized at:

$$F(x) = \frac{1}{2} log \frac{p(y=1|x)}{p(y=-1|x)} \tag{3}$$

Where

$$p(y=1|x) = \frac{e^{F(x)}}{e^{-F(x)} + e^{F(x)}} \tag{4}$$

$$p(y=-1|x) = \frac{e^{-F(x)}}{e^{-F(x)} + e^{F(x)}} \tag{5}$$

Algorithm 4.

3. The proposed method

Variable Selection Ensemble Machine Learning Algorithm (VSEMLA) comprises four different ensemble machine learning algorithms

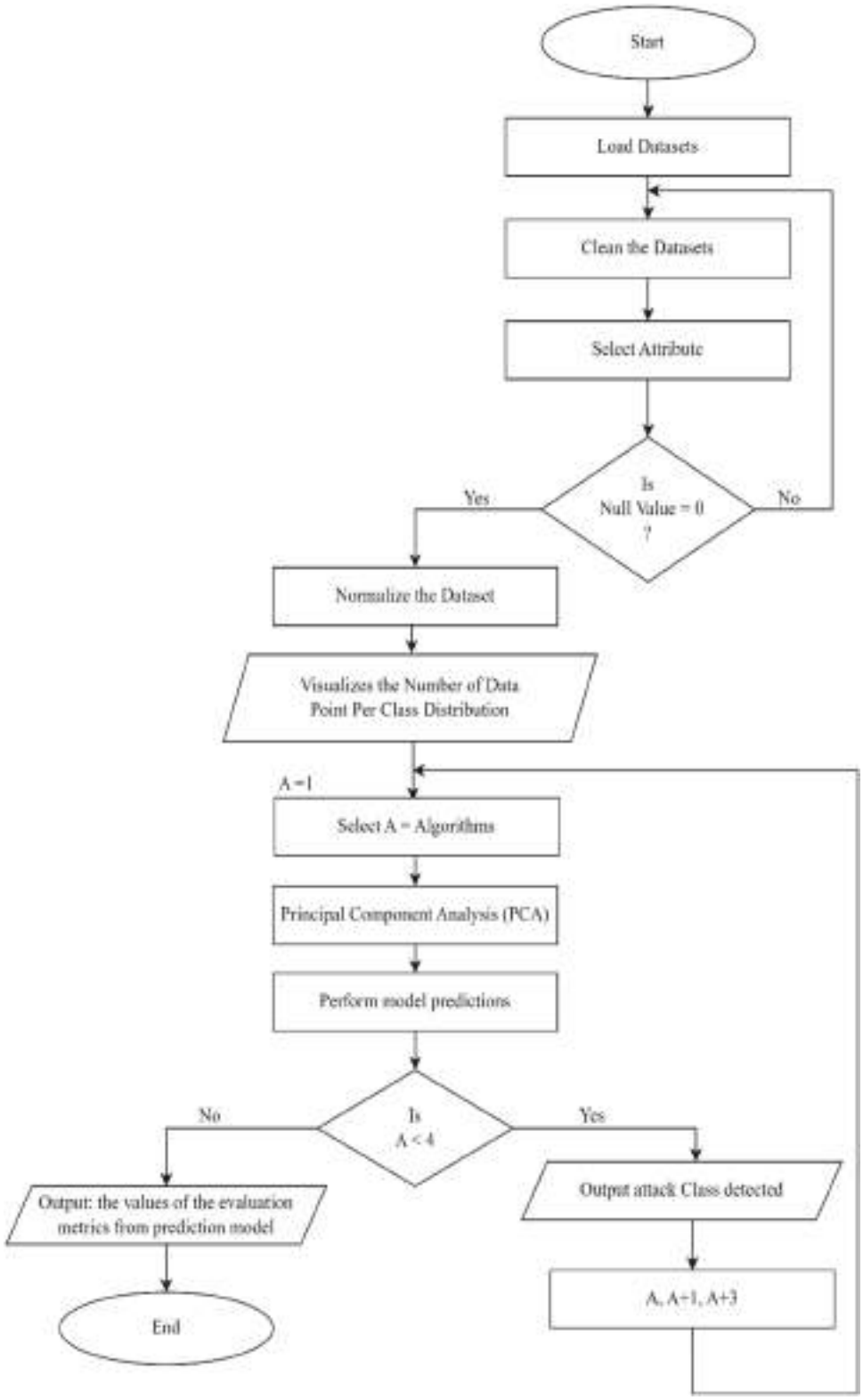

Fig. 2. Flowchart Diagram of the proposed IDS Model.

"Bagging, RandomForest, AdaBoost and LogitBoost". The Bagging and RandomForest algorithms employed the Bootstrap technique for the supervised training of the model by considering several base models trained separately in parallel forms, such as the output of the model as the aggregate collection of the mean average. While AdaBoost and LogitBoost algorithms employed supervised training of the model by considering several base-model trains sequentially where the output of the initial model serves as the input of the next model respectively until the whole models are trained, the final output of the models is the output of the trained model. Bagging also takes less time to build the model and produces low false positives. Still, it is very useful for large and high-dimensional data to reduce variance within a noisy dataset. AdaBoost

Algorithm 5
VSEMLA Algorithm.

1. Input: Datasets (WSN-DS, NSL-KDD, UNSW-NB15) for training and testing,
   Feature selection technique (Principal component analysis),
   Classifier algorithms (RandomForest, AdaBoost, Bagging and LogitBoost algorithms);
2. Output: Prediction Models;
3. Begin: Data preprocessing;
   Cleaning;
   Normalization;
4. End;
5. Begin: Feature extraction;
   Use Principal Component Analysis (PCA) to select the features;
6. End;
7. Begin: Classification;
   Train the RandomForest Classifier;
   Train the AdaBoost Classifier;
   Train the Bagging Classifier;
   Train the LogitBoost Classifier;
   Test evaluation on PCA_RandomForest model using the test datasets;
   Test evaluation on PCA_AdaBoost model using the test datasets;
   Test evaluation on PCA_Bagging model using the test datasets;
   Test evaluation on PCA_LogitBoost model using the test datasets;
8. End;
9. Begin: Model Prediction;
   Predict on PCA_RandomForest model using the test datasets;
   Predict on PCA_AdaBoost model using the test datasets;
   Predict on PCA_Bagging model using the test datasets;
   Predict on PCA_LogitBoost model using the test datasets;
10. Return: the prediction model results;

and LogitBoost algorithms minimize the training errors by combining the set of weak learners into strong learners and reducing the rate of false positives in the intrusion detection system model. Thus, VSEMLA leverages both boosting and bootstrapping techniques, with the advantage of low bias from the bootstrapping technique and low variance from the boosting technique. Bagging is used on weak learners with high variance and low bias (Luo, 2022; Ngo et al., 2022), RandomForest is used on weak learners with low variance and low bias (Han et al., 2021; Mushagalusa et al., 2024; Pellagatti et al., 2021), and both AdaBoost and LogitBoost are used on weak learners with low variance and high bias (Lahmiri et al., 2020; Sui & Ghosh, 2024), thus giving the advantage to VSEMLA algorithm to result in an intrusion detection system models with high performance, low variance and low bias since it combines the bagging algorithm, the random forest algorithm, and the LogitBoost algorithm in a parallel fashion, as shown in the architecture of the proposed method in Fig. 1, such that each predictive model is evaluated in parallel. The final prediction is selected based on the desired application of the model. Moreover, PCA's linear nature might not capture all the intricate, non-linear patterns in the data. This limitation could impact the performance, especially in complex network traffic datasets. The combination of PCA with ensemble methods can be computationally demanding. Boosting algorithms like AdaBoost and LogitBoost involve multiple iterations, which can be resource-intensive, especially with large datasets. AdaBoost, in particular, is sensitive to noisy data and outliers, which can affect overall performance and lead to higher false alarm rates.

Fig. 2 depicts the operational flowchart of the proposed method's Variable Selection Ensemble Machine Learning Algorithm (VSEMLA). It begins by loading and cleaning the datasets and then proceeds by picking the attack attribute. Before normalization and checking for null values to guarantee the cleaning phase is completed, or else it would be repeated until every duplicate and null value is removed from the datasets. It was then visualized to see the number of data points per class distribution or the relationship among the attributes. A counter for the ensemble machine learning algorithms is set, and principal component analysis (PCA) is selected with each algorithm to predict each model in a parallel fashion. It's then checked for the number of chosen algorithms for each alteration. When it is less than four, it saves the prediction output. It instructs the counter to select the following algorithm to perform the next model prediction, output the attack classes detected for the whole iteration, and end the process. Algorithm 5 depicts the process that is involved.

3.1. Datasets

An attack dataset reflects the real-world attack scenarios from the laboratory's simulated cyberattack experiments (Sahu et al., 2021). In the experimental evaluation, three separate datasets, WSN-DS, NSL-KDD, and UNSW-NB15 datasets were employed:

The WSN-DS is a specialized dataset for detecting four types of DoS attacks in a wireless sensor network, specifically Cluster-Based Wireless Sensor Networks (CBWSN): blackhole, flooding, grayhole, and scheduling attacks, all of which are referred to as energy depletion attacks. Almomani et al. (2016) created the dataset in a Network Simulation Two (NS2) environment with 100 nodes in a 10,000-square-meter region. It resulted in eighteen attributes of a class label of around 374,661 data records for intrusion detection systems in wireless sensor networks. The dataset can be used to prevent infiltration by prohibiting malicious nodes from entering the network, with DoS attacks being the most hazardous and damaging on WSNs due to vulnerabilities to security threats.

The most commonly used dataset for analyzing network internet traffic is the NSL-KDD dataset, and the KDD Cup was a 1999 international knowledge discovery and data mining tools competition to gather traffic data (Imrana et al., 2022). The competition aimed to develop a network intrusion detection model that can be used to differentiate malicious network connections from Normal traffic. As more than just a direct consequence, a large volume of internet traffic data was collected and bundled into the KDD-99 data set, and the NSL-KDD was brought in from the University of New Brunswick as the cleaned-up version (Roy et al., 2022). The dataset contains four types of attacks that an anomaly intrusion detection system can detect: Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L).

The UNSW-NB15 dataset contains 45 features (Alsumaini, 2023), three of which are character categorical (proto, service, and state) and ten attack class labels: DoS, worms, exploits, analysis, generic, shellcode, reconnaissance, fuzzers, backdoors, and Normal. According to Bagui et al. (2019); Zhang et al. (2022a), the UNSW-NB15-NB15 dataset was

Table 1
Models' Comparison on WSN-DS Dataset.

Model	TPR	FPR	Precision	Recall	F-Score	ROC Area	Acc	Time (*sec*)
PCA_RandomForest	100	0.0	100	100	100	100	100	133.3
PCA_AdaBoost	92.2	1.4	92.2	92.2	92.2	98.4	92.3	6.5
PCA_Bagging	99.8	1.4	99.8	99.8	99.8	99.9	99.8	84.2
PCA_LogitBoost	98.9	6.3	98.9	98.9	98.9	99.4	98.9	88.5

created with IXIA Perfect Storm by the Australian Centre for Cybersecurity, is a network-based dataset that captures modern traffic patterns and low-footprint intrusions.

3.2. Performance evaluation metrics

The following metric parameters evaluate the performance indicators:

Time is taken, Accuracy, True Positive Rate (TPR), False Positive Rate (FPR), Recall score, precision score, F1 score and the area under the curve (AUC)-Receiver operation characteristic (ROC), which helps in visualizing the performance of the classifier by giving the best estimate of the classifier's performance on the model. Also, a confusion matrix is used to visualise the model performance better.

The indicators evaluation consists of several component matrices identified below as follows:

TN = True negative, which signifies correctly predicted as Normal.

FN = False negative, which signifies mis-predicted as Normal.

TP = True positive, which signifies correctly predicted as abnormal.

FP = False positive, which signifies mis-predicted as abnormal.

The above parameters are combined to form different equations, the metrics primarily used in all research-related works in the literature review. These equations are the evaluation indicators used for the experiment to select the accuracy, precision, recall score, F_1 score, the area under the curve (AUC) of the receive operational characteristics (ROC), TPR, FPR and the model-built time.

The accuracy is the percentage of the sample data that have been correctly detected as normal and abnormal data, as shown in Eq. (6):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

The precision is the percentage of the correctly predicted data out of the total data expected to be abnormal behaviour. Thus, a high precision indicates how lower the error rate of the algorithm used in the model for normal behaviour of the data as shown in Eq. (7):

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

The recall score is the percentage of the abnormal behaviour correctly predicted out of the total abnormal data, as shown in Eq. (8). Thus, it means that when the value of the recall score is higher, then it indicates that the model has a meagre mis-detection rate for abnormal behaviour.

$$Recall\ Score = \frac{TP}{TP + FN} \tag{8}$$

The F1 Score is the harmonic multiplication of the precision with the recall score, which indicates the quality of the model performance as shown in Eq. (9):

$$F1\ Score = \frac{2 * Precision * Recall\ score}{Precision + Recall\ Score} \tag{9}$$

The false positive rate is the percentage of the number of Normal traffic flows predicted as an intrusion from the Normal traffic flows as expressed in Eq. (10):

$$False\ Positive\ Rate\ (False\ Alarm\ Rate) = \frac{FP}{FP + TN} \tag{10}$$

The True positive rate is the percentage of the number of attacks flows predicted correctly as attacks from the total number of attack traffic flows in the dataset and is expressed as shown in Eq. (11):

$$True\ Positive\ Rate(Detection\ Rate) = \frac{TP}{TP + FN} \tag{11}$$

4. Result analysis and discussion

The experimental results were implemented concurrently on the WEKA simulator and the Python 3.7 notebook in Anaconda software, all

Fig. 3. PCA_RandomForest_WSN-DS.

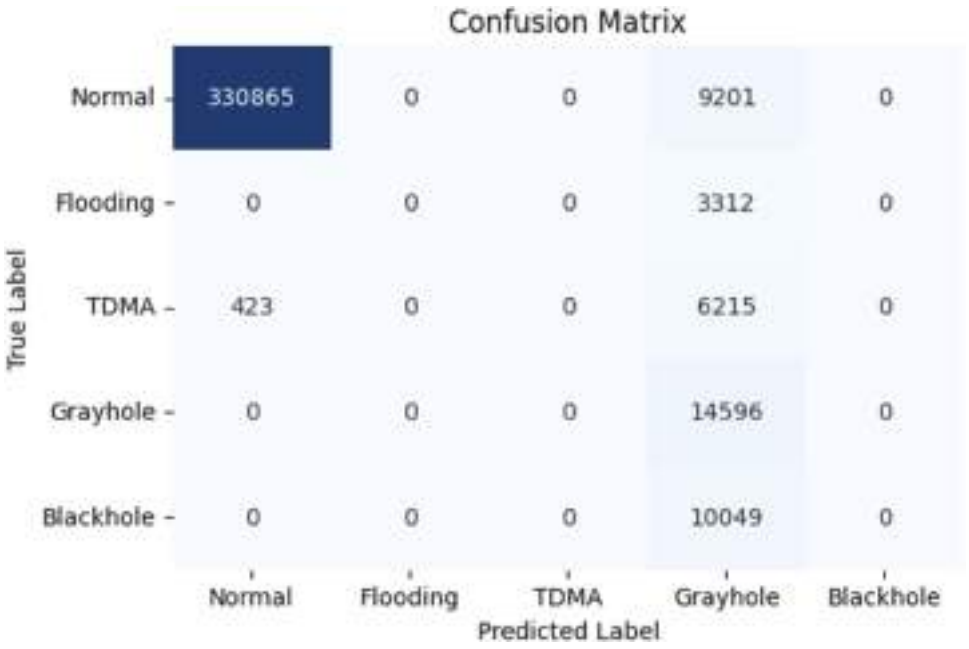

Fig. 4. PCA_AdaBoost_WSN-DS.

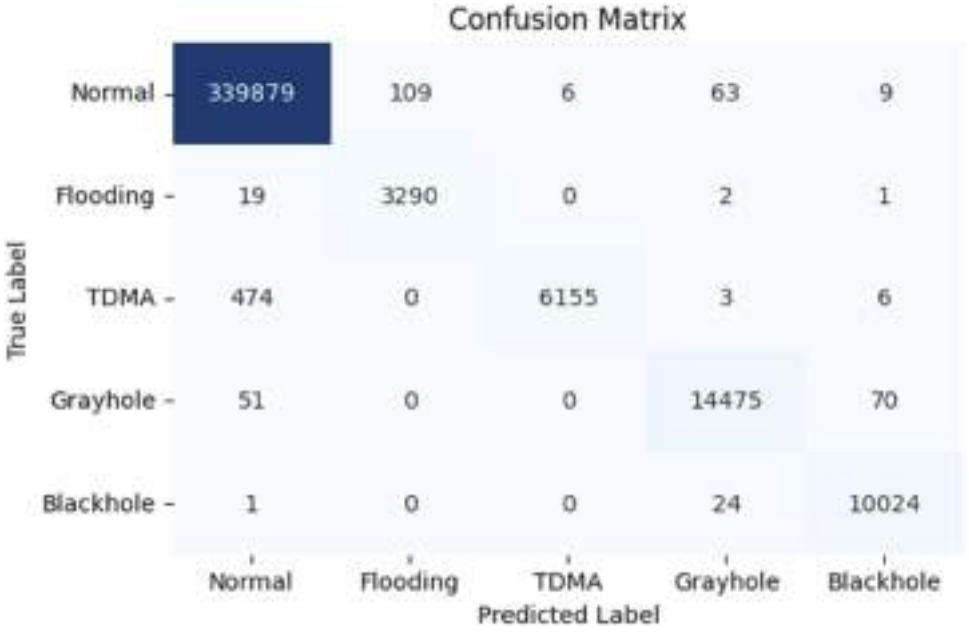

Fig. 5. PCA_Bagging_WSN-DS.

open-source. A principal component analysis (PCA) was used for feature selection, and four algorithms, RandomForest, AdaBoost, Bagging, and LogitBoost, were used as classifiers to build the intrusion detection system models, which were evaluated on three different datasets: the WSN-DS, NSL-KDD, and UNSW-NB15 datasets, using various evaluation metrics. The findings of the experiments were studied, and the discussion of the analysis is offered below:

4.1. Models comparison on different datasets

Table 1 compares the model to the WSN-DS dataset and shows that it performs ideally based on the indicator assessment measures and is well-suited for the task it was designed for. Based on the model prediction results depicted in the confusion matrix shown in Figs. 3–6, PCA_RandomForest has perfect attack detection without bias, PCA_-Bagging and PCA_LogitBoost have a low model bias, and PCA_AdaBoost has a high model bias by detecting the majority of the attack classes as grayhole attacks. Fig. 7 depicts the models' ROC-AUC curves, indicating improved model performance. The confusion matrix thoroughly examines how each model categorizes various attack types. The PCA_R-andomForest model is completely unbiased in its attack detection. For a security-focused application where misclassifying attacks could have terrible repercussions, it accurately classifies all attack types. With low model bias, the PCA_Bagging and PCA_LogitBoost models perform well overall but may still contain a few errors or misclassifications. The PCA_AdaBoost model exhibits a high degree of model bias, especially when identifying most attack classes as grayhole attacks. It suggests a serious problem with the model's performance, as it tends to incorrectly

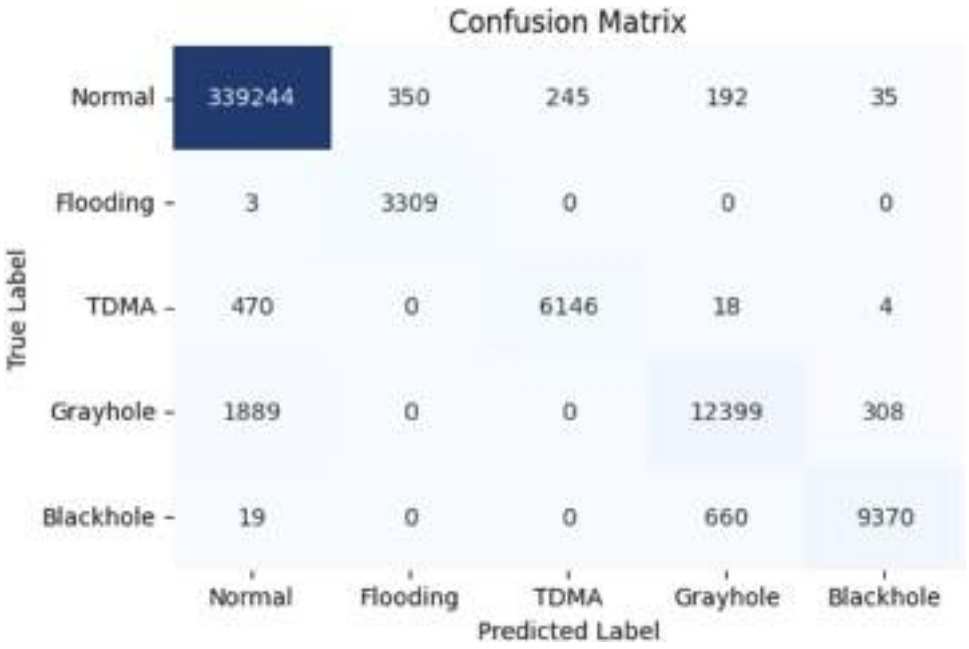

Fig. 6. PCA_LogitBoost_WSN-DS.

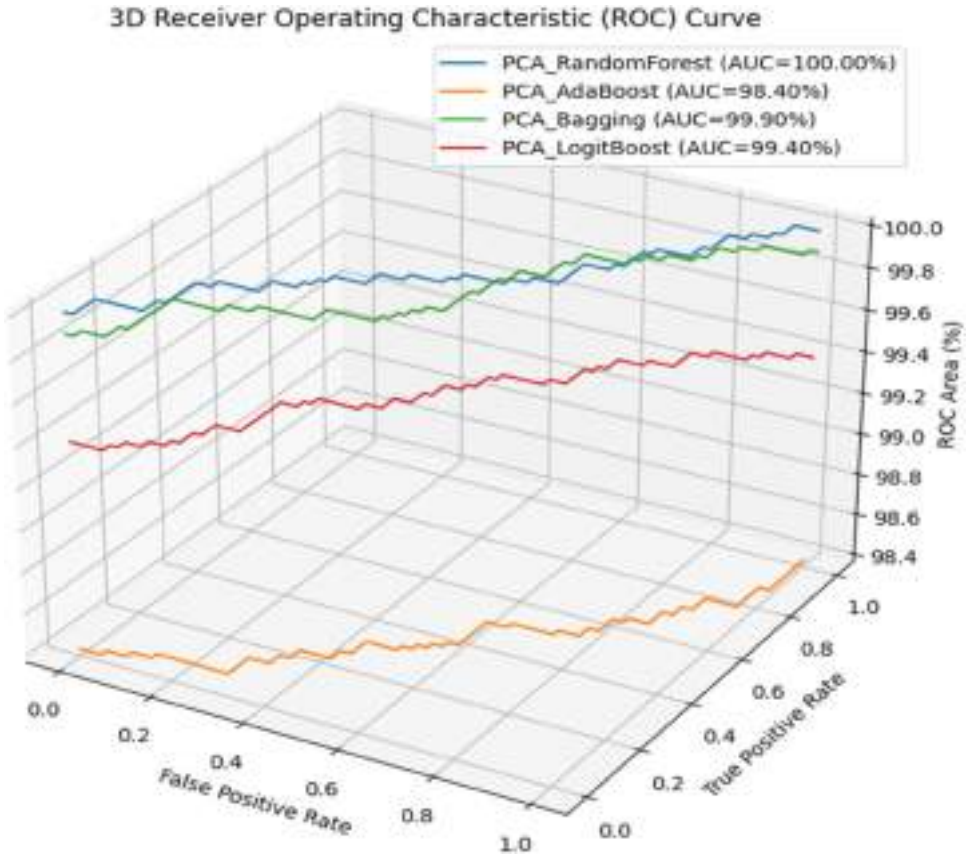

Fig. 7. ROC_AUC of the model Comparison on WSN-DS Dataset.

group different attack types into a single category, which lowers the model's overall efficacy and reliability. The ROC-AUC curves demonstrate the models' capacity to differentiate between various attack types. Better model performance in terms of true positive rate versus false positive rate is thus indicated by the enhanced ROC-AUC score.

The model's performance on the NSL-KDD dataset is shown in Table 2, Fig. 8, Fig. 9, and Fig. 10 of the confusion matrix, which shows a perfect model prediction with no bias. In contrast, Fig. 11 shows a missed detection of some of the attack classes, indicating the presence of model biasing in the prediction model. However, Fig. 12 of the ROC-AUC curve showed improved model performance. The models accurately categorize all instances into their respective categories without

Table 2
Models' Comparison on NSL-KDD Dataset.

Model	TPR	FPR	Precision	Recall	F-Score	ROC Area	Acc	Time (*sec*)
PCA_RandomForest	100	0.0	100	100	100	100	100	15.0
PCA_AdaBoost	89	9.5	89	89	89	92.4	89.0	1.6
PCA_Bagging	100	0.0	100	100	100	100	100	13.2
PCA_LogitBoost	100	0.0	100	100	100	100	100	76.4

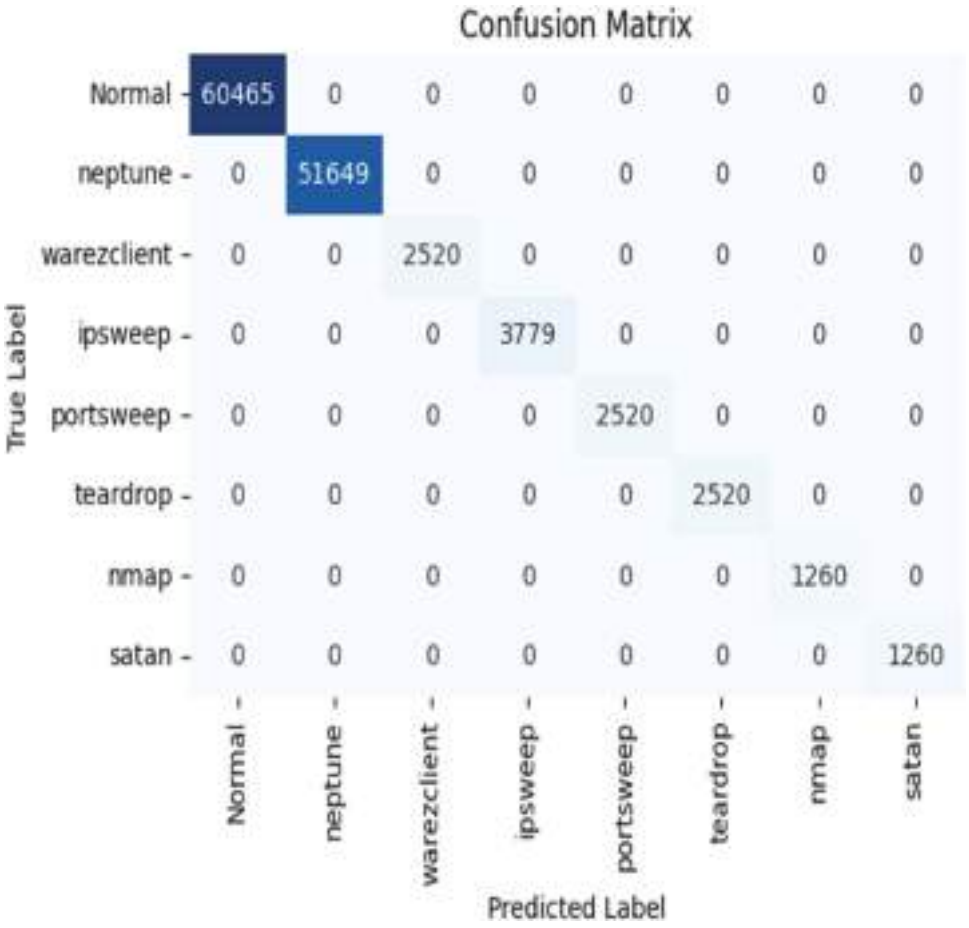

Fig. 8. PCA_RandomForest_NSL-KDD.

misclassification, as demonstrated by the confusion matrix, which reveals perfect predictions free from bias. Thus, except for the second model, which suggests certain biases, it performs extraordinarily well in accurately recognizing the Normal and attack classes. Bias can majorly affect a security system's efficacy by permitting some attacks to go unnoticed and potentially dangerous. The models' strong performance in distinguishing between legitimate and malicious traffic is indicated by the high ROC-AUC value, which promotes more accurate and dependable detection.

Table 3 shows how the model compares to the UNSW-NB15 dataset. Fig. 13 of the confusion matrix depicts a flawless model prediction with no bias. In contrast, Fig. 14 recognizes all attack types as generic attacks, indicating the presence of substantial model bias, while Figs. 15 and 16 show a model prediction with low bias. Fig. 17 depicts the ROC-AUC curve for model prediction, indicating more excellent model performance. In an ideal situation where the model is exceptionally dependable, the first confusion matrix displays perfect model prediction with no bias, showing that the model correctly classifies all instances into their appropriate categories. Significant model bias is evident from the second confusion matrix, which categorizes all attack types as generic attacks. Such bias reduces the model's effectiveness and dependability because it cannot distinguish between various attack types. The final two models, however, display low bias model predictions, indicating that although they may still have a few small misclassifications, overall performance is good, pointing to a more balanced and trustworthy model. The models' excellent capacity to accurately identify threats is indicated by their high ROC-AUC value, which can lead to more effective security measures.

4.2. Comparison of the models on the three data sets

The PCA_RandomForest model outperformed the other two on all three datasets (WSN-DS, NSL-KDD, and UNSW-NB15), as shown in Table 4 and Fig. 18. On the WSN-DS, NSL-KDD, and UNSW-NB15 datasets, PCA_RandomForest significantly outperforms other models, indicating its superior robustness, efficiency, and detection capabilities. Because of this, it's a precious intrusion detection approach that offers improved security, dependability, and affordability while safeguarding network settings.

The PCA_AdaBoost model performed poorly across all datasets, indicating a significant bias in model prediction, as seen in Table 5 and Fig. 19. It combines Principal Component Analysis (PCA) with AdaBoost

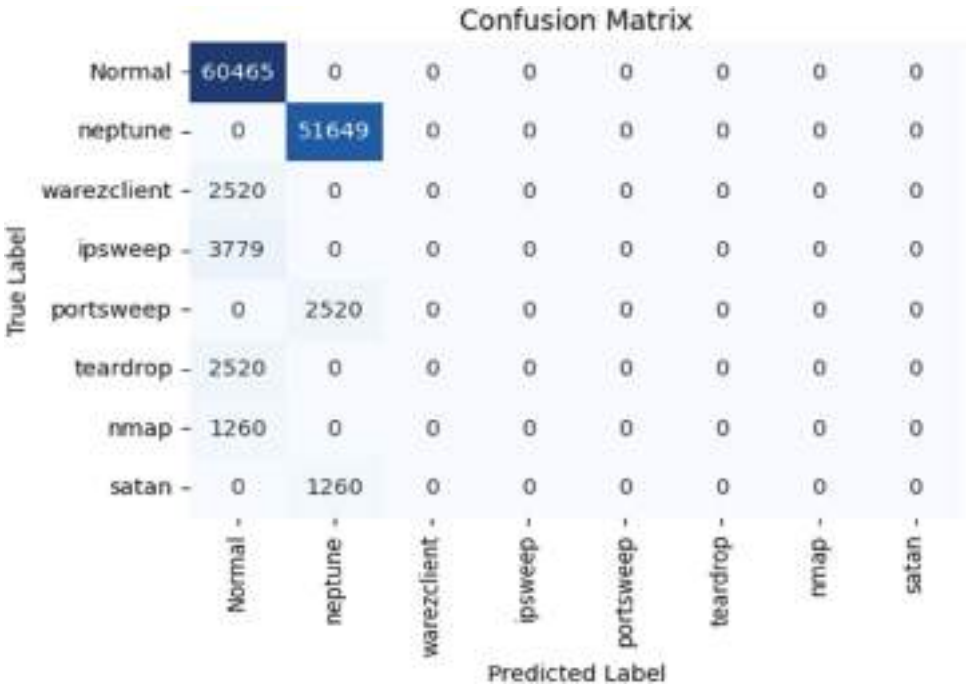

Fig. 9. PCA_AdaBoost_NSL-KDD.

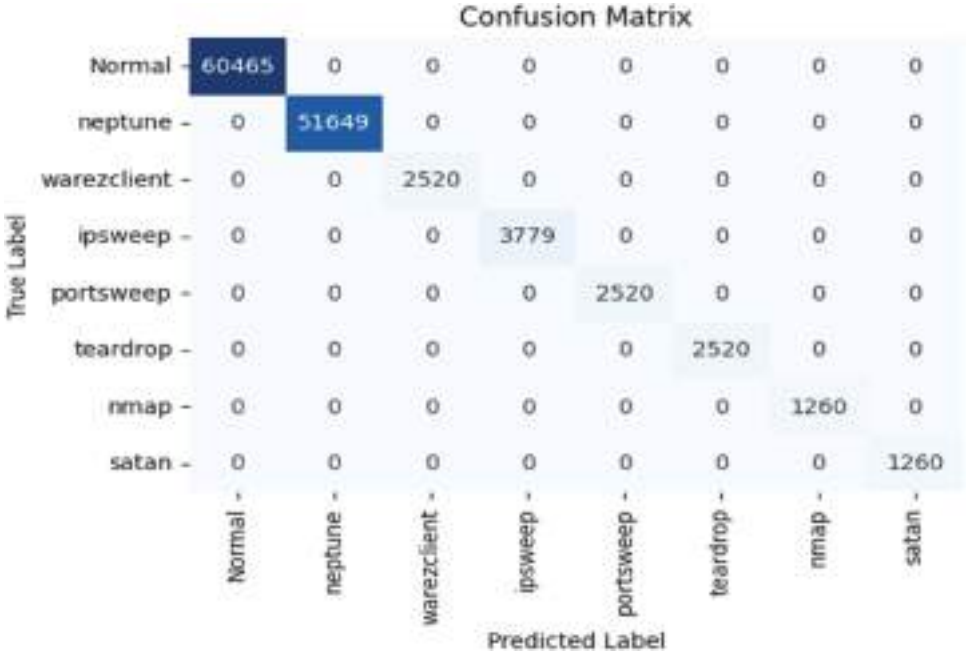

Fig. 10. PCA_Bagging_NSL-KDD.

(Adaptive Boosting) but did not achieve good results when tested on multiple datasets. Poor performance generally means that the model's predictions were inaccurate or had high error rates.

The PCA_Bagging model performs better on the WSN-DS and NSL-KDD datasets, and the UNSW-NB15 dataset demonstrates a low bias in the model, as shown in Table 6 and Fig. 20. PCA reduces the dimensionality of the data by breaking it down into a collection of orthogonal components. In contrast, Bagging is an ensemble technique that combines the predictions of several models trained on various subsets of the data to increase the stability and accuracy of machine learning algorithms. The PCA_Bagging model shows low bias for the UNSW-NB15 dataset. It indicates no consistent under- or overestimation of one direction in the model's predictions. Low bias is a good thing since it means the model accurately identifies the patterns in the data without swerving, of course. Based on its low bias on the UNSW-NB15 dataset, the PCA_Bagging model appears to have solid predictions, signifying that it could produce superior results on some datasets.

The PCA_LogitBoost model performed well on the WSN-DS and NSL-KDD datasets but poorly on the UNSW-NB15 datasets, as illustrated in Table 7 and Fig. 21 of the model prediction results. The LogitBoost ensemble boosting method fits multiple logistic regression models to the data, sequentially adjusting the weights of incorrectly predicted

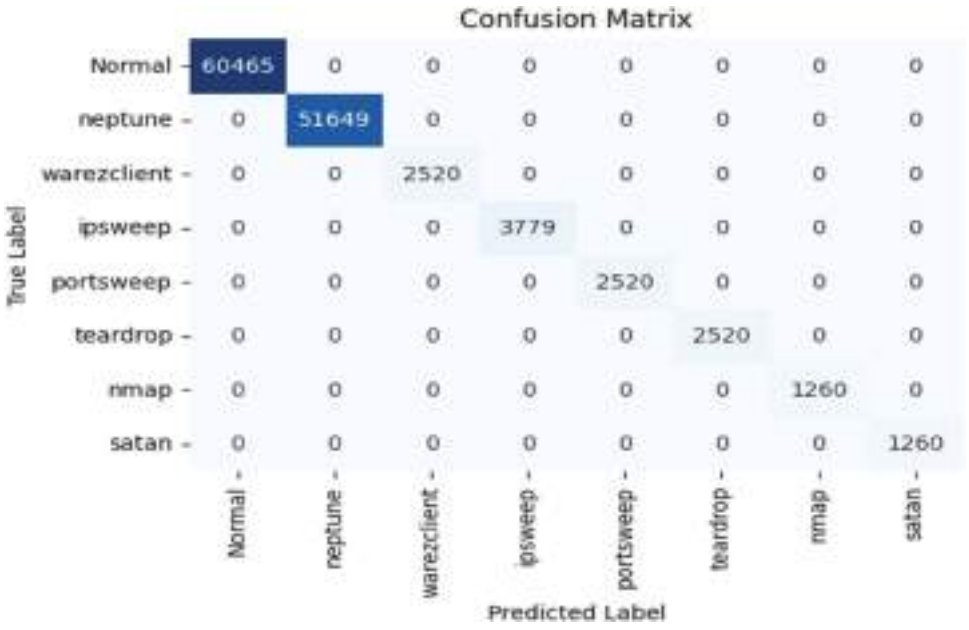

Fig. 11. PCA_LogitBoost_NSL-KDD.

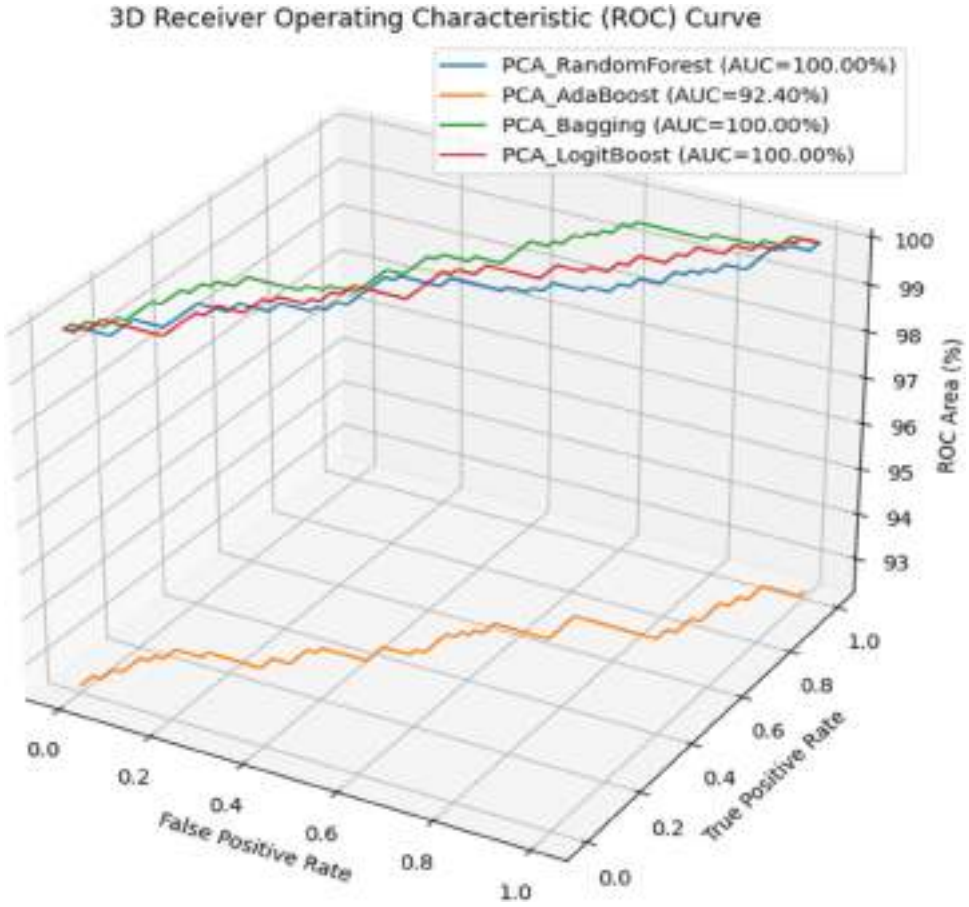

Fig. 12. ROC_AUC of the model Comparison on NSL-KDD Dataset.

instances to improve the model's accuracy.

4.3. Comparison of the models execution time on the three data sets

Table 8 and Fig. 22 compare the model's execution duration across different datasets. The PCA_RandomForest model has the longest execution time on the WSN-DS dataset, the PCA_AdaBoost model has the shortest execution time on all three datasets, and the PCA_LogitBoost model has the fastest on the NSL-KDD and UNSW-NB15 datasets. However, for the NSL-KDD dataset, the PCA_AdaBoost model takes the

Table 3
Models' Comparison on UNSW-NB15 Dataset.

Model	TPR	FPR	Precision	Recall	F-Score	ROC Area	Acc	Time (*sec*)
PCA_RandomForest	100	0.0	100	100	100	100	100	25.1
PCA_AdaBoost	67.9	9.6	67.9	67.9	67.9	87.1	67.9	2.0
PCA_Bagging	93.4	0.8	93.4	93.4	93.4	99.7	93.4	11.1
PCA_LogitBoost	88.7	1.3	88.7	88.7	88.7	99	88.7	49.4

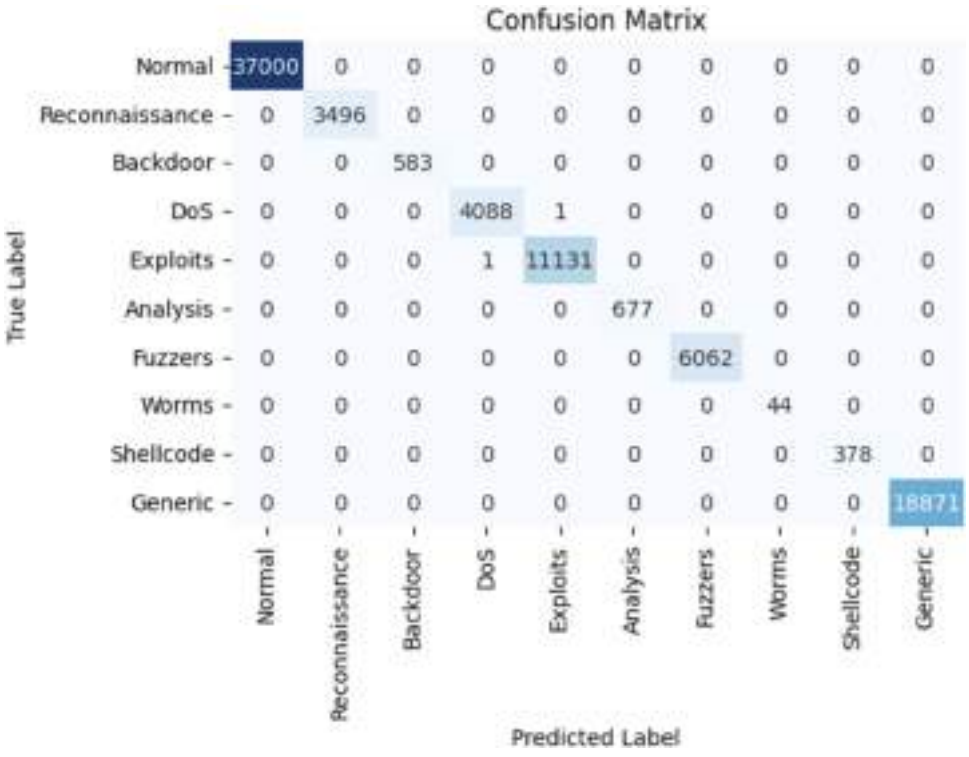

Fig. 13. PCA_RandomForest_UNSW-NB15.

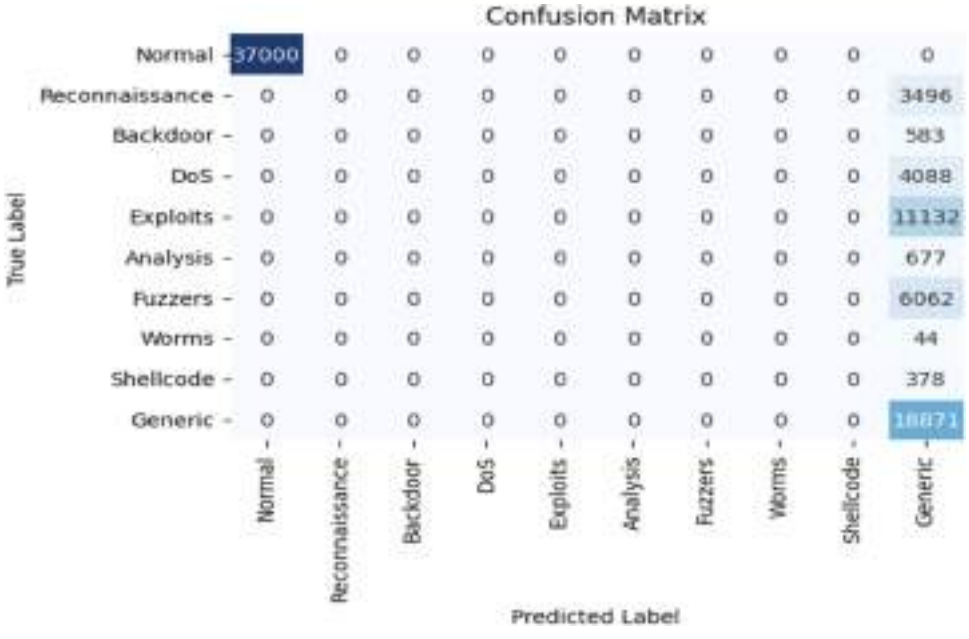

Fig. 14. PCA_AdaBoost_ UNSW-NB15.

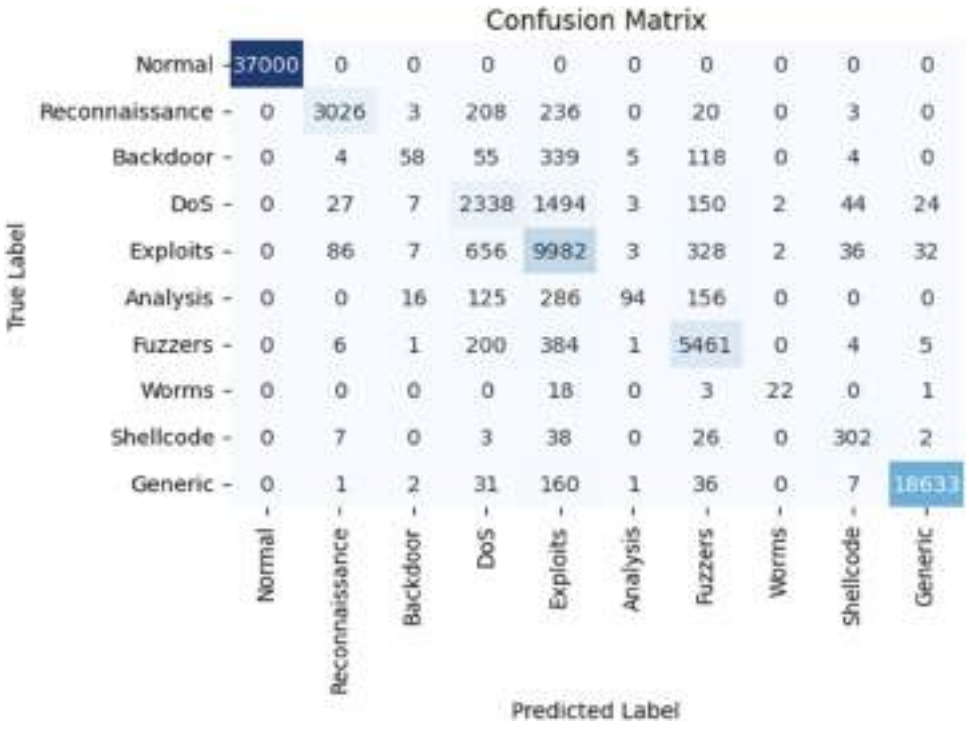

Fig. 15. PCA_Bagging_ UNSW-NB15.

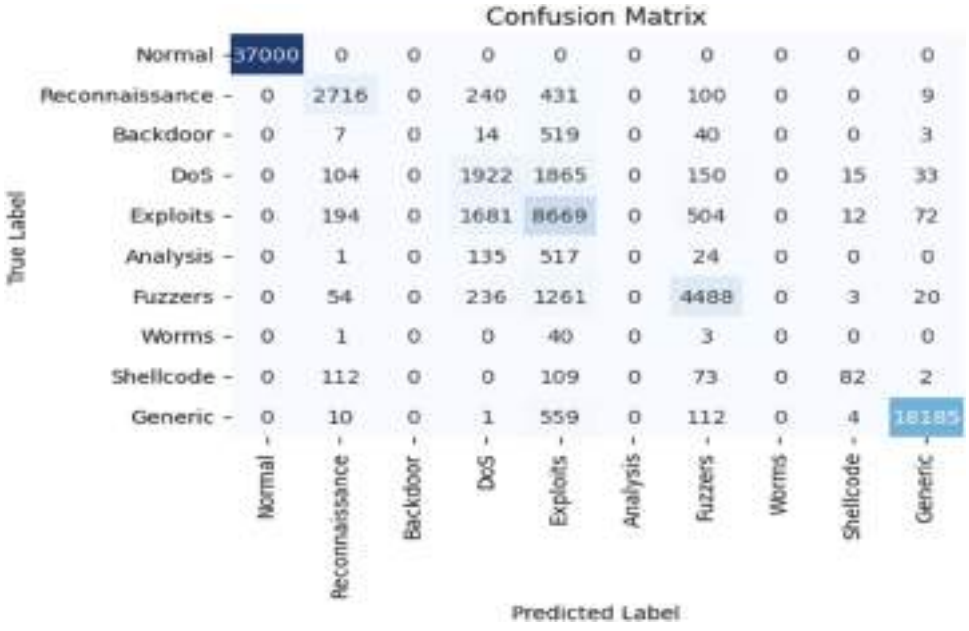

Fig. 16. PCA_LogitBoost_ UNSW-NB15.

shortest execution time.

5. Conclusion

In this research paper, an intrusion detection system (IDS) model was developed using principal component analysis (PCA) for feature selection, and a variable Selection Ensemble Machine Learning Algorithm was used as the proposed method. PCA is paired with the AdaBoost ensemble machine learning technique, which uses stagewise additive modelling to compensate for PCA's deficiency in feature selection in network data by reducing the exponential loss function. Secondly, PCA is used for feature selection, and a LogitBoost classifier technique was used for multiclass classification. It functions as an additive tree regression to compensate for the PCA's deficit by minimizing the logistic loss to offer an optimal classifier output. They were finally implementing Random Forest's low-variance ability, which leverages the bagging strategy to eliminate overfittings. The models were evaluated on three network traffic benchmark datasets: the WSN-DS, NSL-KDD, and UNSW-N15 datasets. The performance of PCA with LogitBoost outperformed that of PCA with AdaBoost for all three datasets used. Thus, PCA's weakness was minimized by the logistic loss of the LogitBoost classifier and less by the exponential loss function of the AdaBoost classifier

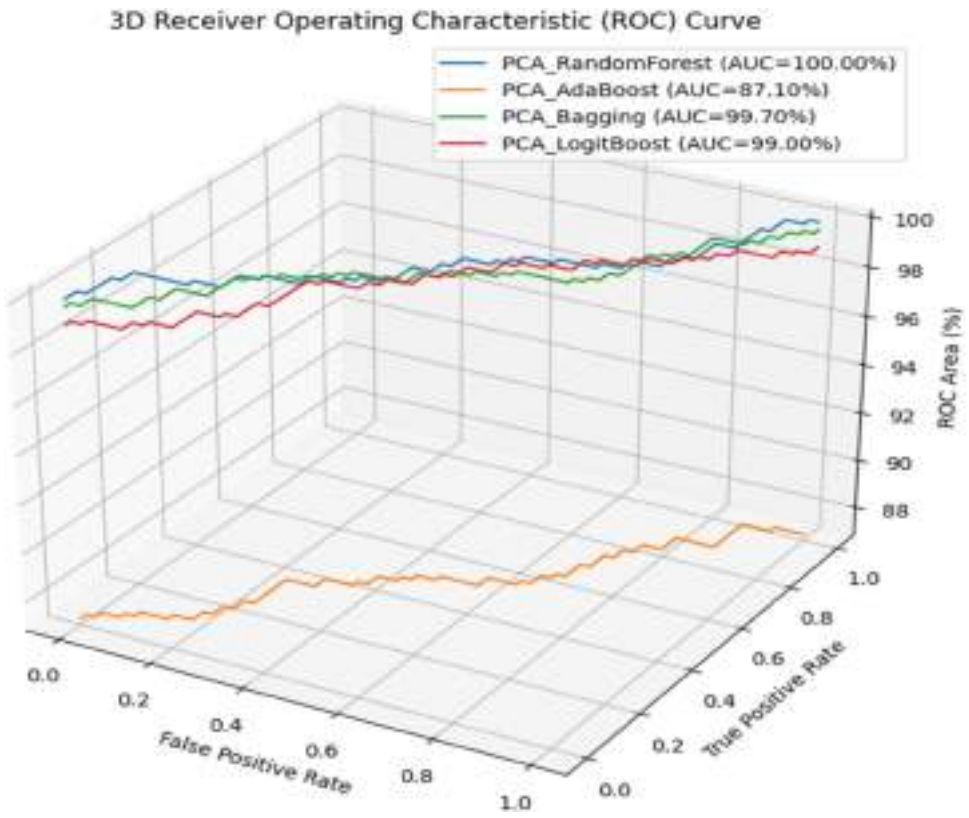

Fig. 17. ROC_AUC of the model Comparison on UNSW-NB15 Dataset.

Table 4
PCA_RandomForest Model Comparison:.

Dataset	Accuracy	Precision	Recall	F-Score
WSN-DS	100	100	100	100
NSL-KDD	100	100	100	100
UNSW-NB15	100	100	100	100

Table 5
PCA_AdaBoost Model Comparison:.

Dataset	Accuracy	Precision	Recall	F-Score
WSN-DS	92.3	92.3	92.2	92.2
NSL-KDD	89.0	89.0	89.0	89.0
UNSW-NB15	67.9	67.9	67.9	67.9

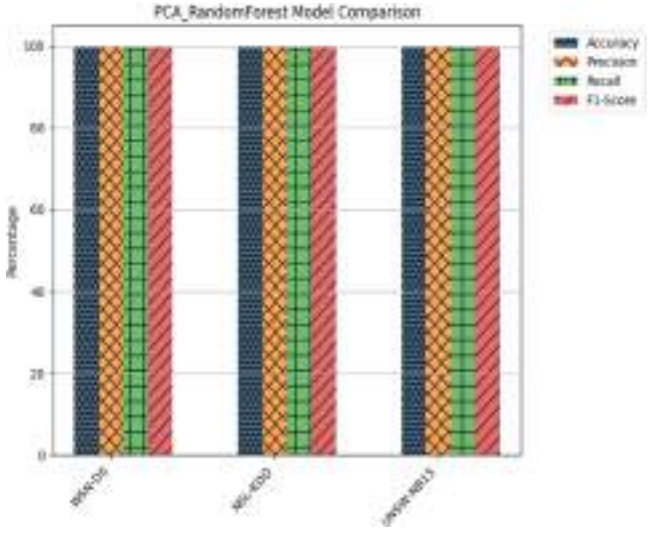

Fig. 18. PCA_RandomForest Model Comparison.

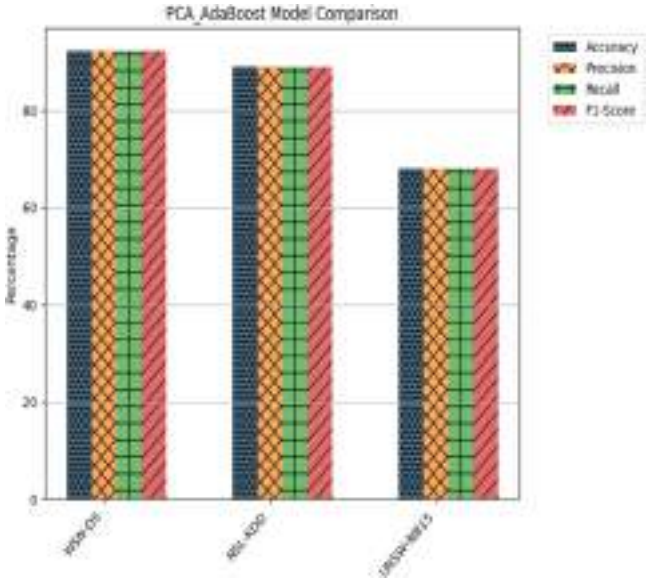

Fig. 19. PCA_AdaBoost Model Comparison.

Ensemble Machine Learning Algorithm used. The bagged estimator has a lower variance than the original estimate, resulting in a significant variance reduction that produces a low bias in the models, and the

Table 6
PCA_Bagging Model Comparison:.

Dataset	Accuracy	Precision	Recall	F-Score
WSN-DS	99.8	99.8	99.8	99.8
NSL-KDD	100	100	100	100
UNSW-NB15	93.4	93.4	93.4	93.4

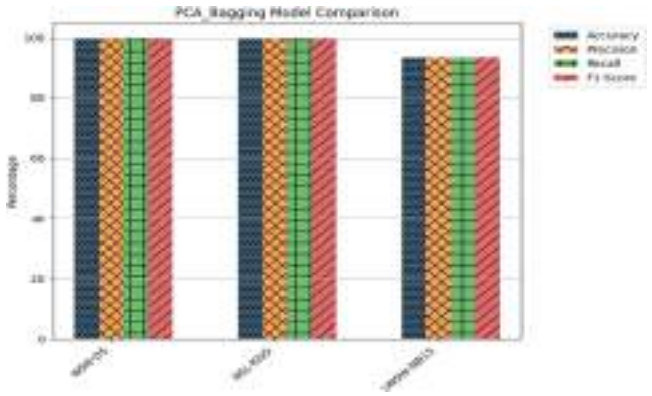

Fig. 20. PCA_Bagging Model Comparison.

Table 7
PCA_LogitBoost Model Comparison:.

Dataset	Accuracy	Precision	Recall	F-Score
WSN-DS	98.9	98.9	98.9	98.9
NSL-KDD	100	100	100	100
UNSW-NB15	88.7	88.7	88.7	88.7

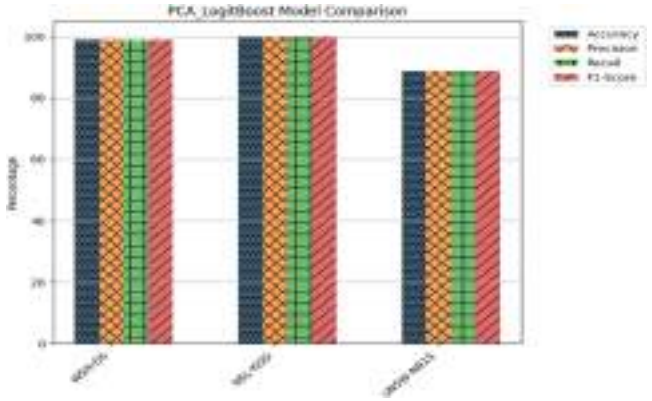

Fig. 21. PCA_LogitBoost Model Comparison.

Table 8
Comparison of Model execution time on different datasets.

Model	Execution Time (*sec*) on WSN-DS	Execution Time (*sec*) on NSL-KDD	Execution Time (*sec*) on UNSW-NB15
PCA_RandomForest	133.3	15.0	25.1
PCA_AdaBoost	6.5	1.6	2.0
PCA_Bagging	84.2	13.2	11.1
PCA_LogitBoost	88.5	76.4	49.4

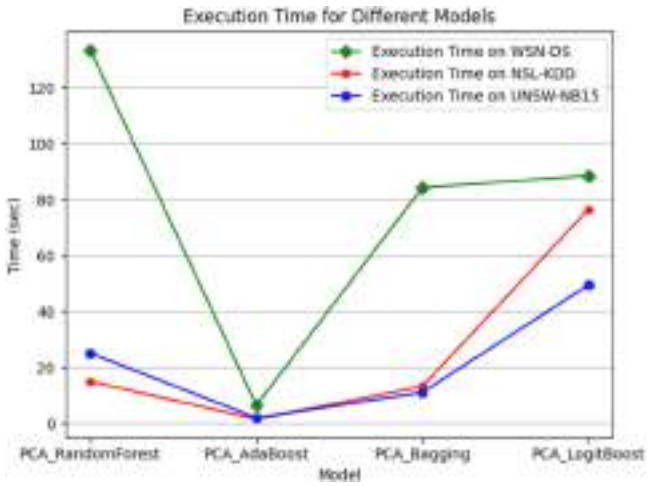

Fig. 22. Comparison of Model Execution Time on Different Datasets.

RandomForest classifier, though slower to compute, eliminates the overfitting problem. Further work is to develop an efficient network security framework with adequate detection of intrusion attacks.

Credit author statement

A.J. drafted the original manuscript by contributing to the methodology, conceptual framework and software development of the research work. At the same time, Dr. I.F.B.I, Dr. S.H.H.M, and Dr. F.B.M supervised the development, validated the models, and reviewed and edited the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The datasets will be available upon request to the corresponding author.

Acknowledgement

On behalf of the postgraduate student, the authors appreciate the Nigerian Petroleum Technology Development Fund (PTDF) agency for providing the student with the scholarship to pursue studies in this research field. A heartfelt appreciation goes to the Universiti Teknologi Malaysia (UTM) for the grant support with reference No: PY/2024/01535 and for providing a convenient research platform for undertaking this research study.

References

Abdelwahed, N. M., El-Tawel, G. S., & Makhlouf, M. (2022). Effective hybrid feature selection using different bootstrap enhances cancers classification performance. *BioData Mining, 15*(1), 24.

Abdoli, M., Akbari, M., & Shahrabi, J. (2023). Bagging supervised autoencoder classifier for credit scoring. *Expert Systems With Applications, 213*, Article 118991.

Al-Fawa'reh, M., Al-Fayoumi, M., Nashwan, S., & Fraihat, S. (2022). Cyber threat intelligence using PCA-DNN model to detect abnormal network behavior. *Egyptian Informatics Journal, 23*(2), 173–185.

Al-Janabi, M., Ismail, M. A., & Ali, A. H. (2021). Intrusion detection systems, issues, challenges, and needs. *Int. J. Comput. Intell. Syst., 14*(1), 560–571.

Alabdulmohsin, I. (2019). Axiomatic characterization of adaboost and the multiplicative weight update procedure. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018*. September 10–14, 2018, Proceedings, Part I 18.

Almomani, I., Al-Kasasbeh, B., & Al-Akhras, M. (2016). WSN-DS: A dataset for intrusion detection systems in wireless sensor networks. *Journal of Sensors, 2016*.
Alsumaini, A.Y.M. (2023). *Two-stage ensemble learning for nids multiclass classification* Hamad Bin Khalifa University (Qatar)].
Ashiku, L., & Dagli, C. (2021). Network intrusion detection system using deep learning. *Procedia Computer Science, 185*, 239–247.
Asselman, A., Khaldi, M., & Aammou, S. (2023). Enhancing the prediction of student performance based on the machine learning XGBoost algorithm. *Interactive Learning Environments, 31*(6), 3360–3379.
Awotunde, J. B., & Misra, S. (2022). Feature extraction and artificial intelligence-based intrusion detection model for a secure internet of things networks. *Illumination of artificial intelligence in cybersecurity and forensics* (pp. 21–44). Springer.
Bagui, S., Kalaimannan, E., Bagui, S., Nandi, D., & Pinto, A. (2019). Using machine learning techniques to identify rare cyber-attacks on the UNSW-NB15 dataset. *Security and Privacy, 2*(6), e91.
Bakır, R., Orak, C., & Yüksel, A. (2024). Optimizing hydrogen evolution prediction: A unified approach using random forests, lightGBM, and Bagging Regressor ensemble model. *International Journal of Hydrogen Energy, 67*, 101–110.
Bao, F., Wu, Y., Li, Z., Li, Y., Liu, L., & Chen, G. (2020). Effect improved for high-dimensional and unbalanced data anomaly detection model based on KNN-SMOTE-LSTM. *Complexity, 2020*(1), Article 9084704.
Barrow, D., Kourentzes, N., Sandberg, R., & Niklewski, J. (2020). Automatic robust estimation for exponential smoothing: Perspectives from statistics and machine learning. *Expert Systems With Applications, 160*, Article 113637.
Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review, 54*, 1937–1967.
Camacho, J., Therón, R., García-Giménez, J. M., Maciá-Fernández, G., & García-Teodoro, P. (2019). Group-wise principal component analysis for exploratory intrusion detection. *IEEE access : practical innovations, open solutions, 7*, 113081–113093.
Chen, Y., Wang, Y., Gu, Y., He, X., Ghamisi, P., & Jia, X. (2019). Deep learning ensemble for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 12*(6), 1882–1897.
Chu, J., Lee, T.-H., & Ullah, A. (2020). Component-wise AdaBoost algorithms for high-dimensional binary classification and class probability prediction. In *Handbook of statistics, 42* pp. 81–114). Elsevier.
Di Mauro, M., Galatro, G., Fortino, G., & Liotta, A. (2021). Supervised feature selection techniques in network intrusion detection: A critical review. *Engineering Applications of Artificial Intelligence, 101*, Article 104216.
Ebenezer, V., Devassy, R., & Kathrine, G. J. W. (2023). Intrusion detection and prevention system to analyse and prevent malware using machine learning. In *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*.
Gajewski, M., Batalla, J. M., Mastorakis, G., & Mavromoustakis, C. X. (2019). A distributed IDS architecture model for smart home systems. *Cluster Computing, 22*, 1739–1749.
Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., & Suganthan, P. N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence, 115*, Article 105151.
Gassais, R., Ezzati-Jivan, N., Fernandez, J. M., Aloise, D., & Dagenais, M. R. (2020). Multi-level host-based intrusion detection system for Internet of things. *Journal of Cloud Computing, 9*(1), 62.
González, S., García, S., Del Ser, J., Rokach, L., & Herrera, F. (2020). A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. *Information Fusion, 64*, 205–237.
Guarascio, M., Cassavia, N., Pisani, F. S., & Manco, G. (2022). Boosting cyber-threat intelligence via collaborative intrusion detection. *Future Generation Computer Systems, 135*, 30–43.
Guezzaz, A., Azrour, M., Benkirane, S., Mohy-Eddine, M., Attou, H., & Douiba, M. (2022). A lightweight hybrid intrusion detection framework using machine learning for edge-based IIoT security. *Int Arab J Inf Technol, 19*(5).
Han, S., Williamson, B. D., & Fong, Y. (2021). Improving random forest predictions in small datasets from two-phase sampling designs. *BMC Medical Informatics and Decision Making, 21*, 1–9.
Hillebrand, E., Lukas, M., & Wei, W. (2021). Bagging weak predictors. *International Journal of Forecasting, 37*(1), 237–254.
Hossain, M. A., & Islam, M. S. (2023a). Ensuring network security with a robust intrusion detection system using ensemble-based machine learning. *Array, 19*, Article 100306.
Hossain, M. A., & Islam, M. S. (2023b). Ensuring network security with a robust intrusion detection system using ensemble-based machine learning. *Array*, Article 100306.
Hu, L., Chen, J., Vaughan, J., Aramideh, S., Yang, H., Wang, K., et al. (2021). Supervised machine learning techniques: An overview with applications to banking. *International Statistical Review, 89*(3), 573–604.
Imrana, Y., Xiang, Y., Ali, L., Abdul-Rauf, Z., Hu, Y.-C., Kadry, S., et al. (2022). χ 2-BidLSTM: A feature driven intrusion detection system based on χ 2 statistical model and bidirectional LSTM. *Sensors, 22*(5), 2018.
Jain, H., Khunteta, A., & Srivastava, S. (2020). Churn prediction in telecommunication using logistic regression and logit boost. *Procedia Computer Science, 167*, 101–112.
James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). Resampling methods. *An introduction to statistical learning: With applications in python* (pp. 201–228). Springer.
Kareem, M. K., Aborisade, O. D., Onashoga, S. A., Sutikno, T., & Olayiwola, O. M. (2023). Efficient model for detecting application layer distributed denial of service attacks. *Bulletin of Electrical Engineering and Informatics, 12*(1), 441–450.
Kazak, E., & Pohlmeier, W. (2023). Bagged pretested portfolio selection. *Journal of Business & Economic Statistics, 41*(4), 1116–1131.
Kizza, J. M. (2024). System intrusion detection and prevention. *Guide to computer network security* (pp. 295–323). Springer.
Kocher, G., & Kumar, G. (2021). Machine learning and deep learning methods for intrusion detection systems: Recent developments and challenges. *Soft Computing, 25*(15), 9731–9763.
Konhäuser, K., Wenninger, S., Werner, T., & Wiethe, C. (2022). Leveraging advanced ensemble models to increase building energy performance prediction accuracy in the residential building sector. *Energy and Buildings, 269*, Article 112242.
Lahmiri, S., Bekiros, S., Giakoumelou, A., & Bezzina, F. (2020). Performance assessment of ensemble learning systems in financial data classification. *Intelligent Systems in Accounting, Finance and Management, 27*(1), 3–9.
Li, T., Kou, G., & Peng, Y. (2020). Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods. *Information Systems, 91*, Article 101494.
Lucchese, L. V., de Oliveira, G. G., & Pedrollo, O. C. (2020). Attribute selection using correlations and principal components for artificial neural networks employment for landslide susceptibility assessment. *Environmental Monitoring and Assessment, 192*(2), 129.
Luo, C. (2022). A comparison analysis for credit scoring using bagging ensembles. *Expert Systems, 39*(2), e12297.
Lv, L., Wang, W., Zhang, Z., & Liu, X. (2020). A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine. *Knowledge-Based Systems, 195*, Article 105648.
Mafarja, M., Thaher, T., Al-Betar, M. A., Too, J., Awadallah, M. A., Abu Doush, I., & Turabieh, H. (2023). Classification framework for faulty-software using enhanced exploratory whale optimizer-based feature selection scheme and random forest ensemble learning. *Applied Intelligence, 53*(15), 18715–18757.
Majidian, Z., TaghipourEivazi, S., Arasteh, B., & Babai, S. (2023). An intrusion detection method to detect denial of service attacks using error-correcting output codes and adaptive neuro-fuzzy inference. *Computers and Electrical Engineering, 106*, Article 108600.
Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2018). A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE Communications Surveys & Tutorials, 21*(1), 686–728.
Mohammed, A., & Kora, R. (2023). A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University-Computer and Information Sciences, 35*(2), 757–774.
Muneer, S., Farooq, U., Athar, A., Ahsan Raza, M., Ghazal, T. M., & Sakib, S. (2024). A critical review of artificial intelligence based approaches in intrusion detection: A Comprehensive analysis. *Journal of Engineering, 2024*(1), Article 3909173.
Mushagalusa, C. A., Fandohan, A. B., & Glèlè Kakaï, R. (2024). Random forest and spatial cross-validation performance in predicting species abundance distributions. *Environmental Systems Research, 13*(1), 23.
Ngo, G., Beard, R., & Chandra, R. (2022). Evolutionary bagging for ensemble learning. *Neurocomputing, 510*, 1–14.
Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A comprehensive evaluation of ensemble learning for stock-market prediction. *Journal of Big Data, 7*(1), 20.
Osho, O., Hong, S., & Kwembe, T. A. (2021). Network intrusion detection system using principal component analysis algorithm and decision tree classifier. In *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*.
Pellagatti, M., Masci, C., Ieva, F., & Paganoni, A. M. (2021). Generalized mixed-effects random forest: A flexible approach to predict university student dropout. *Statistical Analysis and Data Mining: The ASA Data Science Journal, 14*(3), 241–257.
Putra, M. A. R., Ahmad, T., & Hostiadi, D. P. (2023). Dimensional feature reduction for detecting botnet activities. In *2023 25th International Conference on Advanced Communication Technology (ICACT)*.
Rajadurai, H., & Gandhi, U. D. (2021). An empirical model in intrusion detection systems using principal component analysis and deep learning models. *Computational Intelligence, 37*(3), 1111–1124.
Ravi, V., Chaganti, R., & Alazab, M. (2022). Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system. *Computers and Electrical Engineering, 102*, Article 108156.
Remeseiro, B., & Bolon-Canedo, V. (2019). A review of feature selection methods in medical applications. *Computers in biology and medicine, 112*, Article 103375.
Roy, S., Li, J., Choi, B.-J., & Bai, Y. (2022). A lightweight supervised intrusion detection mechanism for IoT networks. *Future Generation Computer Systems, 127*, 276–285.
Sahoo, R., Pasayat, A. K., Bhowmick, B., Fernandes, K., & Tiwari, M. K. (2022). A hybrid ensemble learning-based prediction model to minimise delay in air cargo transport using bagging and stacking. *International Journal of Production Research, 60*(2), 644–660.
Sahu, A., Wlazlo, P., Mao, Z., Huang, H., Goulart, A., Davis, K., & Zonouz, S. (2021). Design and evaluation of a cyber-physical testbed for improving attack resilience of power systems. *IET Cyber-Physical Systems: Theory & Applications, 6*(4), 208–227.
Salman, M., Husna, D., Apriliani, S. G., & Pinem, J. G. (2018). Anomaly based detection analysis for intrusion detection system using big data technique with learning vector quantization (LVQ) and principal component analysis (PCA). In *Proceedings of the 2018 International Conference on Artificial Intelligence and Virtual Reality*.
Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. A. (2020). Performance analysis of machine learning algorithms in intrusion detection system: A review. *Procedia Computer Science, 171*, 1251–1260.
Selvakumar, B., & Muneeswaran, K. (2019). Firefly algorithm based feature selection for network intrusion detection. *Computers & Security, 81*, 148–155.
Singh, A., Nagar, J., Amutha, J., & Sharma, S. (2023). P2CA-GAM-ID: Coupling of probabilistic principal components analysis with generalised additive model to

predict the k− barriers for intrusion detection. *Engineering Applications of Artificial Intelligence, 126*, Article 107137.
Singh, C. E., & Vigila, S. M. C. (2023). Fuzzy based intrusion detection system in MANET. *Measurement: Sensors, 26*, Article 100578.
Sothe, C., De Almeida, C., Schimalski, M., La Rosa, L., Castro, J., Feitosa, R., Dalponte, M., Lima, C., Liesenberg, V., & Miyoshi, G. (2020). Comparative performance of convolutional neural network, weighted and conventional support vector machine and random forest for classifying tree species using hyperspectral and photogrammetric data. *GIScience & Remote Sensing, 57*(3), 369–394.
Sui, Q., & Ghosh, S. K. (2024). Active learning for stacking and AdaBoost-related models. *Stats, 7*(1), 110–137.
Tanha, J., Abdi, Y., Samadi, N., Razzaghi, N., & Asadpour, M. (2020). Boosting methods for multi-class imbalanced data classification: An experimental review. *Journal of Big Data, 7*, 1–47.
Thudumu, S., Branch, P., Jin, J., & Singh, J. (2020). A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data, 7*, 1–30.
Udas, P. B., Karim, M. E., & Roy, K. S. (2022). SPIDER: A shallow PCA based network intrusion detection system with enhanced recurrent neural networks. *Journal of King Saud University-Computer and Information Sciences, 34*(10), 10246–10272.
Uddin, M. P., Mamun, M. A., & Hossain, M. A. (2021). PCA-based feature reduction for hyperspectral remote sensing image classification. *IETE Technical Review, 38*(4), 377–396.
Um, I., Lee, G., & Lee, K. (2023). Adaptive boosting for ordinal target variables using neural networks. *Statistical Analysis and Data Mining: The ASA Data Science Journal, 16*(3), 257–271.
Valavi, R., Elith, J., Lahoz-Monfort, J. J., & Guillera-Arroita, G. (2021). Modelling species presence-only data with random forests. *Ecography, 44*(12), 1731–1742.
Wang, L., Mao, Z., Xuan, H., Ma, T., Hu, C., Chen, J., & You, X. (2022). Status diagnosis and feature tracing of the natural gas pipeline weld based on improved random forest model. *International Journal of Pressure Vessels and Piping, 200*, Article 104821.
Wang, P., Fan, E., & Wang, P. (2021). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters, 141*, 61–67.
Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y., & Han, H. (2022). A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security, 116*, Article 102675.
Zhang, L., Xie, X., Xiao, K., Bai, W., Liu, K., & Dong, P. (2022a). MANomaly: Mutual adversarial networks for semi-supervised anomaly detection. *Information Sciences, 611*, 65–80.
Zhang, T., Han, D., Marino, M. D., Wang, L., & Li, K.-C. (2022b). An evolutionary-based approach for low-complexity intrusion detection in wireless sensor networks. *Wireless Personal Communications*, 1–24.
Zhang, W., Wu, C., Zhong, H., Li, Y., & Wang, L. (2021). Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization. *Geoscience Frontiers, 12*(1), 469–477.
Zhang, Y., Ma, J., Liang, S., Li, X., & Liu, J. (2022c). A stacking ensemble algorithm for improving the biases of forest aboveground biomass estimations from multiple remotely sensed datasets. *GIScience & Remote Sensing, 59*(1), 234–249.
Zhiqiang, L., Mohiuddin, G., Jiangbin, Z., Asim, M., & Sifei, W. (2022). Intrusion detection in wireless sensor network using enhanced empirical based component analysis. *Future Generation Computer Systems, 135*, 181–193.
Zounemat-Kermani, M., Batelaan, O., Fadaee, M., & Hinkelmann, R. (2021). Ensemble machine learning paradigms in hydrology: A review. *Journal of Hydrology, 598*, Article 126266.

Ayuba JOHN is a PhD candidate in Computer Science at Universiti Teknologi Malaysia (UTM). In 2010, he received his B. Eng. Engineering Degree in Computer Engineering from the University of Maiduguri, Nigeria, and in 2017, he received his M.Eng. Computer Engineering Degree from the University of Benin, Nigeria. He worked for the National Control Centre (NCC) in the Transmission Company of Nigeria (TCN) as a transmission engineer in 2014, and he is a member of the Nigerian Society of Engineers (NSE). He is now a lecturer at Nigeria's Federal University Dutse. His research areas of interest include cybersecurity, machine learning and wireless sensor networks. E-mail: ayuba.john@fud.edu.ng or john@graduate.utm.my. https://orcid.org/0000–0003–0496–765x.

Ismail Fauzi ISNIN received PhD and MS degrees in Network System Engineering from the University of Plymouth UK in 2011 and 2004, respectively. He is a senior lecturer at the School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Malaysia. He is a member of the Pervasive Computing Research Group and the School of Computing. His research interests are wired and wireless computer networks, mobile ad-hoc networks and communication, high performance, and parallel computing. He can be contacted at the following e-mail: ismailfauzi@utm.my. https://orcid.org/0000–0002–9765–3491.

Syed Hamid Hussain MADNI is currently based in Malaysia and working as a Senior Lecturer at the School of Electronic and Computer Science, University of Southampton of Engineering, Malaysia. He received his PhD in 2020 from Universiti Teknologi Malaysia (UTM) and worked as a senior lecturer before moving to his current station. His area of research is "Optimal Resource Scheduling for Infrastructure as a Service in Cloud Computing based on Cuckoo Search". He received an MS (CS) degree in 2009 from the Federal Urdu University of Arts, Science and Technology, Islamabad, Pakistan. His areas of interest are cloud computing, algorithm analysis, network security, e-commerce, web development and Internet of Things. He has published about 18 research papers in High Impact Journals. He is also conducting training on research publications at different international universities in various countries. Email: s.h.h.madni@soton.sc.uk or madni4all@yahoo.com https://orcid.org/0000–0002–3816–1382.

Farkhana Binti Muchtar is a Senior Lecturer at the Faculty of Computing, Universiti Teknologi Malaysia (UTM). With a solid foundation in computer networking, she specializes in pervasive computing. She holds a diploma and a bachelor's degree in Computer Science from UTM, as well as an MSc in Computer Science from Universiti Utara Malaysia, and a PhD in Computer Science from UTM. Her research interests include Named Data Networking (NDN), the Internet of Things (IoT), cloud computing, fog and edge computing, and blockchain technologies. She has made significant contributions to her field through various publications in academic journals, conference proceedings, and edited books. Email: farkhana@utm.my https://orcid.org/0000–0002–5636–5741.