Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 3, Issue 9, 2025

MACHINE LEARNING-BASED NETWORK INTRUSION DETECTION USING RANDOM COMMITTEE ENSEMBLES: A MULTI-METRIC PERFORMANCE EVALUATION

Jamal Hussain Arman[*1], **Muhammad Usama Sharaf** [2], **Syed Tahir Ali Shah** [3], **Najamuddin Sohu** [4], **Sindhu** [5], **Urooj Tariq**[6]

[1]*Deparment of Electrical Engineering, National University of Computer and Emerging Sciences, Pakistan*
[2]*Institute of Electrical Electronics and computer Engineering, University of the Punjab*
[3]*TEMA - Centre for Mechanical Technology and Automation, Department of Mechanical Engineering, University of Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal*
[4]*Assistant Professor/ Director HR, Department of Information Technology, GC University, Hyderabad*
[5]*Department of Software Engineering, Muhammad Ali Jinnah University, Karachi, Pakistan*
[6]*Department of computer science, Abbottabad University of Science and Technology, Abbottabad, Pakistan*

[*1]jamalhussainarman901@gmail.com, [2]m.usama.sharaf@gmail.com, [3]syedtahiralishah@ua.pt, [4]Najam_sohu@yahoo.com, [5]sindhuwaghella@gmail.com, [6]uroojeman444@gmail.com

DOI: https://doi.org/10.5281/zenodo.17129656

Keywords
Cyber security
Network intrusion detection system
Random committee
Cross validation
Performance assessment

Article History
Received: 11 June 2025
Accepted: 21 August 2025
Published: 16 September 2025

Copyright @Author

Corresponding Author: *
Jamal Hussain Arman

Abstract

Network intrusion detection systems (NIDSs) are used as a tool to prevent the network from various attacks. Network intruders use the software's flaw to launch a number of assaults against computer network security, which results in the loss of key data. To this end, researchers in the recent past presented various intelligence-based models, each with its strengths and weaknesses. To design a proactive protective system, machine learning (ML) is widely used to monitor and respond to any cyber threats quickly. To keep the discussion of previous studies, this study proposes Random Committee (RC), an ML-based model for NIDS, which is a supervised ML technique and applied to labeled data. The Proposed model results are also compared to a number of Machine Learning models, which include Random Tree, Hoeffding Tree, Decision Stump, Decision Table, K-Nearest Neighbor, Naive Bayes, Boosting, and Bagging using the Waikato Environment for Knowledge Analysis (WEKA) tool. The assessments are made using multiple metrics, which are the Matthew correlation coefficient, false positive rate, true positive rate, accuracy, precision, and receiver operating characteristic area. The Kaggle and NSL-KDD datasets are used in both training and testing. The findings indicate that the proposed models are superior and have an accuracy of 99.9, and this forms a baseline for future studies. In turn, these results are the baseline for the researchers in terms of deciding and setting the priority of cyber-related properties in achieving a successful and best NIDS.

INTRODUCTION

In present days, the usage of information and communication technologies (ICTs) is steadily increasing, and so are cyber-attacks on these systems. To counter anomalous threats with a safe detection system, the research community conducted a variety of studies. Network security researchers and specialists are especially focused on identifying and thwarting cyber-attacks in order to ensure safe communication. [1]. To secure the Confidentiality, Integrity, and Availability (CIA) of the data, numerous companies and organizations spend a large portion of their budgets on network security [2]. In a digitally connected world, it can be challenging to separate beneficial data from large amounts of valuable information, making it harder to spot intruders or dangerous data. In such a challenging environment, Network Intrusion Detection Systems (NIDS) are crucial for protecting the network and lowering risks [3]. Since conventional security measures like firewalls and antivirus software are unable to recognize and thwart emerging threats, machine learning (ML) models are used in the field of network security [4]. Unauthorized access to a network for malicious intentions is known as intrusion. NIDS is a software and hardware system that continuously keeps track of every activity on the network. Despite some major problems like low accuracy and a high false alarm rate, some NIDS operate extremely effectively [5]. Network security is the main focus of researchers who employ firewalls, antivirus programs, and NIDS to ensure safe and secure communication. In this area, using machine learning (ML) and deep learning (DL) is the best option. Artificial algorithms (AI), which are being utilized realistically to anticipate normal and pathological actions, include machine learning (ML) and deep learning (DL) [6]. ML-based models enable the extraction of important data from extremely complicated data and priceless information. Because ML-based NIDS learns from the data pattern itself, it increases accuracy and requires less human knowledge [7]. NIDS's main objectives are to identify, evaluate, and stop any cyberattack on a network's host machine. For effective detection, NIDS often employs anomaly-based, signature-based, or a mix of the two. In order to identify abuse occurring both within and outside of a network, NIDS gathers data from both individual and numerous computers connected to the network [8]. Fig. 1 shows the taxonomy of the NIDS [9].

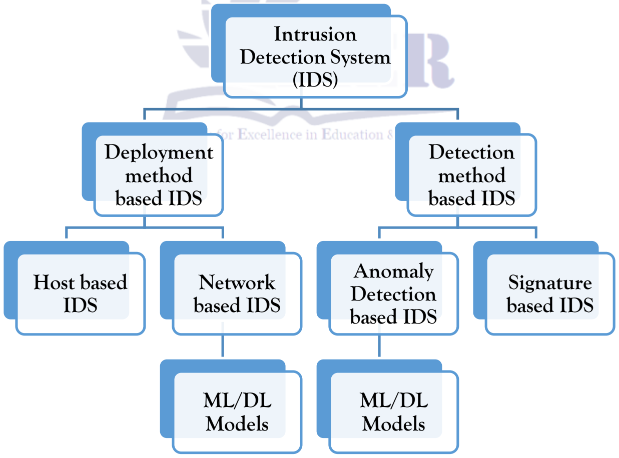

Intrusion Detection System (IDS)
Deployment method based IDS
Detection method based IDS
Host based IDS
Network based IDS
Anomaly Detection based IDS
Signature based IDS
ML/DL Models
ML/DL Models

Figure 1. Intrusion Detection System Model and Machine Learning

Additionally, ICT is used in smart companies, where all devices are wirelessly connected and data can freely flow between all operations. The most efficient, structured, and integrated security models are needed in these sectors [10]. According to research, the cost of cybercrime worldwide will surpass $265 billion (USD) by 2031, indicating that attacks on organizations, networks, and customers will occur every 2 seconds [11]. The cost of cybercrime worldwide was $3 trillion in 2015, but by 2025, it will exceed $10.5 trillion [12].

Keeping all these issues in mind, researchers proposed various models for NIDS each with its advantages and disadvantages. The main contributions made in this work are; firstly, this study proposes Random Committee (RC) based model for NIDS using both binary class and multi class datasets that are NSL-KDD and Kaggle respectively. Secondly, both datasets have high number of attributes which lead to over fitting issue. To overcome this issue, the proposed model reduces the over fitting issue significantly. Thirdly, in previous works, Pareto principle or two different datasets used for training and testing but in this work we have used cross validation (CV) for each dataset. To reduce the over fitting issue more, 10 fold cross validation (FCV) used in this work. Lastly, data is labeled in both datasets and for labeled data analysis we used supervised ML techniques to design an optimal model for NIDS. The proposed model is compared with well-known ML models, which are discussed in the subsequent sections.

Furthermore, Section 2 summarizes previous research done; Section 3 presents the proposed RC model; Section 4 describes the experimental model in detail and Section 5 shows the results and conclusion of the paper.

Literature Review

In recent years, NIDS using machine learning algorithms have gained popularity as a research area. Anomaly and signature-based NIDS, or a mix of the two, are among the many methods used. The literature has a large number of research publications about signature-based NIDS.

The authors evaluate the effectiveness of 48 anomaly detection systems (AIDS) based on ML with the help of a CICIDS2017 dataset. Supervised and unsupervised paradigms are discussed; moreover, the detection rate in the former paradigm is 99.28 %, and detection rate in unsupervised paradigm reaches 60.06 %. The results promote a careful practice in selecting the features and suggest designing the deep-learning models to analyze them thoroughly. The findings show that artificial-neural-network (ANN) architecture provides a better output and the use of K-mean gives the least optimal output. Finally, the paper reports that incorporation of feature-selection method into ML-based AIDS can lead to significantly improved accuracy of AIDS [13].

This paper compared the use of two algorithms in machine-learning namely, artificial neural networks (ANN) and k-nearest neighbors (KNN) with that of NIDS and their accuracies were measured in terms of accuracy, precision, true positive rate (TPR), and false positive rate (FPR). The KNN model has attained 0.9957 accuracy, 0.9949 precision, 0.9959 TPR, and 0.9956 TNR, compared with 0.9923, 0.9910 and 0.9926, 0.9920 in the ANN model. However, the very size of feature selection and the time density calculations are the main limitations, and both algorithms exhibit different performance on the available databases. As such, KNN works on particular data [14].

The authors explored intrusion detection using CICIDS2017 and ISCXIDS2012. A hybrid system which involves a combination of a packed and session classifier reaches the accuracy of 99.8 percent on CICIDS2017 and 97.37 percent on ISCXIDS2012. The model compares with random forest, Adaboosted decision trees, deep neural networks, SMOTE+RF, support vector machines, TSE including rotational forest, extreme learning machines and gradient boosting trees. The major drawback of the method is its high cost and complexity although it is very accurate [15].

The authors compared data-driven Intrusion Detection Systems (IDS) that were implemented using a 10-fold cross validation (FCV) system in the KDDcup99 set of data. They trained 2 models of ML such as Random Forest (RF) and Decision Tree (DT) and the performance metrics included the accuracy, precision, and recall. The RF scored on accuracy, precision and recall 94 % against 93 % of the DT, 99 % against 98 % and 93 % against 92 % [16].

The authors create 1D, 2D, and 3D convolutional neural networks (CNN) for detecting anomalies in the networks. The dataset is partitioned using the Pareto theory or 80/20 rule which states that 20% of the dataset should be used for testing and 80% of the dataset should be used for training. For four classes, all CNN-based models produce good accuracy. Normal, Scan, Theft, and DoS were all detected at a rate of 99.90%, 99.91%, 98.10%, and 99.96%, respectively. FPR was 0.05%, and FNR was 0.67%. The CNN1D model had a minimal recognition rate of 99.74%, CNN2D had a minimal recognition rate of 99.42%, and CNN3D had a minimal recognition rate of 99.03% [17].

The authors propose two ML algorithms Random Forest (RF) and Deep Feed Forward (DFF) classifiers for the classification of data. Different datasets are used and each dataset is divided into 70:30 training and testing set. Both classifiers achieve high detection accuracy by increasing data rate and lowering the false alarm rate. DFF classifier achieve accuracy for NF-

CSE-CIC-IDS2018-v2, CSE-CIC-IDS2018, NF-ToN-IoT-v2, CIC-ToN-IoT and CIC-BoT-IoT are 99.24%, 97.05%, 94.74%, 93.80% and 96.01% respectively. RF classifier achieve accuracy for NF-CSE-CIC-IDS2018-v2, CSE-CIC-IDS2018, NF-ToN-IoT-v2, CIC-ToN-IoT, NF-BoT-IoT-v2 and CIC-BoT-IoT are 99.47%, 98.01%, 99.66%, 99.33%, 100.00% and 98.24% respectively [18].

A leading theme of modern network security research is the characterisation and detection of malicious activity. In this regard, the ContikiNG operating system has been implemented to categorise the NSL-KDD dataset and as a result brought out newer classification of threats which include denial-of-service (DoS) and probe attacks. Also in the same setup, different machine-learning algorithms; decision trees (DT), bootstrap aggregating trees (Bagging), random forests (RF), NARVE Bayes, and AdaBoost were used to apply to the labelled data. The experimental results denote that DT, Bagging, and RF have a high accuracy of classification: 0.966, 0.968, and 0.967, writing respectively. In comparison Naive Bayes and AdaBoost show relatively poor performance with accuracy rates of 0.948 and 0.951 respectively. RF, DT, Bagging and SVM have a precision and F-score of 0.969; 0.969; 0.968; 0.966; 0.967; 0.967; 0.968; 0.957; 0.968; 0.968; 0.968; 0.957; 0.951; 0.957; 0.957; and 0.95 [19].

In this study, the writers design a tree-like model for decision-making and select specific feature selection based on their ranks and scores. The data is divided into small subsets for easy feature encoding and scaling using Python programming language, which lower the model variance and over-fitting issues. The proposed model reduces the complexity and increases the prediction rate for unknown attacks too. Finally, the proposed IntrDTree model is found to be more efficient than the traditional models having 0.98 precision, 0.981 recall, 0.98 F-score and 0.981 accuracy [20].

The publishers used the NSL-KDD dataset consisting of 42 features with10 continuous, 6 binary, 23 discrete and 3 categorical features. The dataset contains two classes: normal and attack. In this paper, ReLU activation is used which contains 1024, 768 and 512 neurons in 3 hidden layers. The proposed model accuracy, precision, recall and F1 are 0.824, 0.964, 0.713 and 0.820 respectively. With a training accuracy of 0.9823 for the training set and a testing accuracy of 0.7950, BRCG is a very effective tool for extracting rules from data. This indicates that by using only these rules, one may obtain results on test data that are around 80% correct. [21].

An integrated classification based IDS is presented by the authors and evaluates the performance on both offline and online data sets. UNSW-NB15 dataset which includes multiple attacks such as: DoS, Exploit, Normal, Probe and Generic are used and the results are compared to other existing traditional DT based models, the value of numerous evaluation criteria (e.g. MFM = 84.5%, ADR = 90.32, FAR = 2.01 %, etc.) has a greater performance. To prevent these kinds of attacks, a signature is added to the recommended IDS model. In order to assess the efficacy of the suggested model, this study also creates a real-time data set (RTNITP18) at the NIT Patna CSE lab. The accuracy of the suggested model is 83.8% [22].

The experimenter implements the proposed strategy into action and extracts the key features from the raw flow data. Code of this strategy is written in Python, with the ML algorithms implemented using PyTorch. The dataset's train, validation, and test splits are 75 %, 15 % and 10 %, respectively. A limitation is imposed during the training phase to limit the huge number of samples to use per class, while no adjustment is made during the testing phase to reduce the class imbalance. The result shows that the accuracy, precision and data rate are 99.66%, 99.65% and 99.66 respectively for the UNSW-NB15 dataset while the accuracy, data rate and precision are 99.56%, 99.56 and 99.56 respectively for CICIDS2017 [23].

In this study, ANN based model with wrapper feature selection is presented and compared with the SVM model. For classification, trial and error methods apply to the NSL-KDD dataset. The results show that the ANN with wrapper feature selection performs best with a detection rate of 94.02%. ANN model accuracy for 17 features is 94.02% and for 35 features is 83.68% while for SVM model the accuracy for 17 features is 81.78% and for 35 features is 82.34% [24].

Previous research works have two major limitations. Firstly, authors used only one performance metric that is accuracy to measure to efficiency of the model on a single dataset only and secondly, authors did not use new ML technique. Most of the previous work focused on binary classified datasets and neglected multiclass real time data sets. To overcome these problems, this research work used new supervised ML technique i.e. random committee (RC) and analyzed model efficiency using multiple performance metrics to conclude the model efficiency on two different datasets including binary classed and multiclass.

Proposed Machine Learning Model

This section discusses the proposed model, RC for NIDS.

Random Committee

In this technique, several base classifiers are constructed using a variety of seed values chosen at random. This enhances the result by combining the results of classifiers with different seed values and lowering the chance of error [25]. By building many base classifiers using various random number seed values, the final classification result is calculated by averaging the predictions given by each basis classifier [26].

RC is the most powerful algorithm that handles binary, numeric, nominal and missing classes' values and it is a class for building an ensemble of randomizable base classifiers. RC is a novel supervised ML technique which is used for classification. Our selected datasets contained labeled class data and for this purpose we proposed this supervised ML technique which is designed for labeled data classification.

Experimental Setup

This study presents an analysis of the performance of two different datasets that were downloaded from the Kaggle repository to presents ML classification methods for IDS. In Fig. 2, the entire study process is illustrated. Following the selection of datasets, each dataset undergoes a preprocessing phase with the dual objectives of restoring missing values and changing the class attribute from a numerical to a categorical value. After all, the results of applied ML approaches to each dataset are evaluated using different metrics to demonstrate the superior performance of a certain technique.

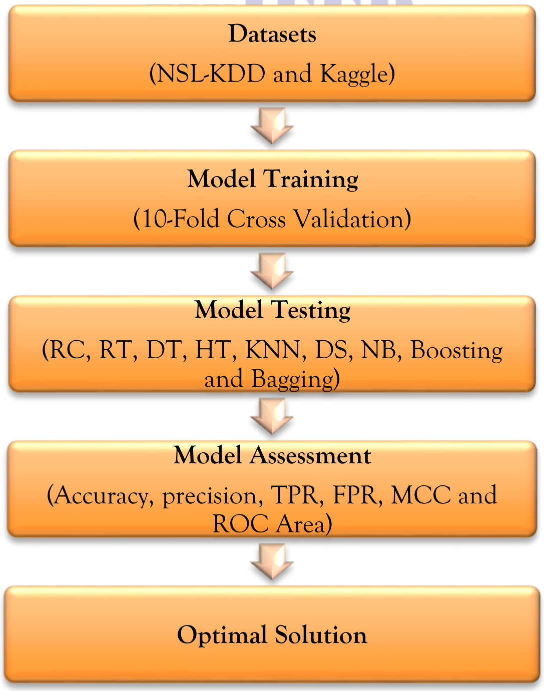

Datasets
(NSL-KDD and Kaggle)
Model Training
(10-Fold Cross Validation)
Model Testing
(RC, RT, DT, HT, KNN, DS, NB, Boosting and Bagging)
Model Assessment
(Accuracy, precision, TPR, FPR, MCC and ROC Area)
Optimal Solution

Figure 2. Flow Chart of the Proposed Study

Furthermore in this section, two datasets: NSL-KDD and Kaggle, 10 FCV, performance metrics: accuracy, precision, TPR, FPR, MCC, ROC area and applied techniques such as: RC, RT, DT, HT, KNN, DS, NB, boosting and bagging are presented.

Datasets

This section presents both binary class datasets and multiclass datasets such as NSL-KDD and Kaggle. These datasets are used in research pertaining to cyber security. Each dataset is composed of a known class label and some attributes. While the overall number of attributes and instances varies, each dataset contain numerical data.

Kaggle Dataset

The Kaggle dataset's samples are divided into five groups: normal, DoS, r21, probe, and u2r. The 125973 instances in this modified dataset—which no longer contains any raw network data—have been divided into two sets for training and testing, as shown in Table 1 [27]. This dataset, which contains several features and instance sets, comprises both typical assaults and some of the more common ones from real-time networking. This dataset represents the most reliable one to assess the actual performance of IDS [8].

NSL-KDD Dataset

While assessing the effectiveness of NIDS in research, the NSL-KDD dataset is often used as a reference. This dataset is an updated version of KDD Cup 99 that incorporates numerous novel attacks. The elimination of redundant and duplicate records in this collection is its key benefit [28]. NSL-KDD is binary classified and designated as normal and anomalous, and it has 41 features [29]. The training set and testing set of this dataset, which has about 125000 attributes total, are separated into two groups as indicated in Table 1 [30].

Table 1. Datasets Statistics

	Training	Testing
Kaggle	113376	12597
NSL-KDD	113376	12597

10 Fold Cross Validation

The dataset is divided into training and testing sets using cross-validation. Before being split up into K groups, the data is combined. The training model receives vast amounts of data, but the testing model receives very little. In the past, researchers employed the 80/20 rule or the percentage split approach; however, in this work, 10 FCV is applied. Fig. 3 illustrates the cross-validation process.

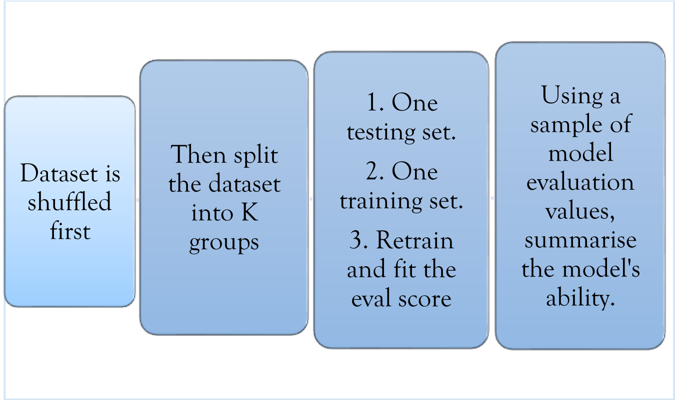

Dataset is shuffled first
Then split the dataset into K groups
1. One testing set.
2. One training set.
3. Retrain and fit the eval score
Using a sample of model evaluation values, summarise the model's ability.

Figure 3. Cross-validation Flowchart

Performance Assessments

The effectiveness of a model is assessed using metrics that measure its performance using a representation generally referred to as a confusion matrix. The effectiveness of classification models is determined by the confusion matrix. Positive (P) and negative (N) binary classification classes are employed in this study for each experiment. The confusion matrix generates 4 outcomes based on the two classes, as shown in Table 2 [31].

Table 2. Representation of Classification Model Results in the Confusion Matrix

Actual Class	Predicted Class	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Indicators of the positive and negative classes are called True Positive (TP) and True Negative (TN), respectively. A false positive (FP) occurs when the model predicts P but the actual response is N. A false negative (FN) occurs when the model predicts N but the correct response is P [32].

The actual class and predicted class values are compared using the confusion matrix, a common paradigm for performance evaluation. Accuracy is defined as the proportion of properly categorized samples to all tested instances. The precision of an object is determined by dividing its true positive values by its total positive values. False positive rate (FPR) is the ratio of misclassified negative samples to all negative samples, whereas true positive rate (TPR) is the ratio of all really detected items to all true samples. The abscissa for each point is FPR, and the ordinate is TPR, indicating the classifier's trade-off between TP and FP. Mathews Correlation Coefficient (MCC) is a balanced statistical index that takes into account both SN and SP, although it is sensitive to the distribution of classes in a testing set. The calculated value is close to 1 indicating that an accurate classification is made [33]. Table3 shows the mathematical forms of all the performance metrics used in this paper.

Table 3. Performance Assessment Metrics Mathematical Form

Performance Metric	Mathematical form	
Accuracy	$\frac{TP + TN}{TP + TN + FN + FP}$	(1)
Precision	$\frac{TP}{TP + FP}$	(2)
TPR	$\frac{TP}{TP + FN}$	(3)
FPR	$\frac{FP}{TN + FP}$	(4)
MCC	$\frac{((TP * TN) - (FP * FN))}{\sqrt{((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))}}$	(5)

Techniques Applied for Comparative Analysis

The RC-based model is compared with some latest ML models that are briefly discussed subsequently.

Random Tree

A decision tree technique called Random Tree (RT) uses N randomly chosen variables at each node to fit several decision trees to a dataset. Each tree is evenly sampled due to the symmetrical distribution of these sets of random decision trees. By combining these evenly spaced trees, RT is created, leading to a more precise and useful model. RT is a hybrid method of a single tree based on random forest (RF) principles, with k randomly selected characteristics at each node in the tree. Consequently, random forests outperform a single tree in terms of accuracy. RT creates a sample of each class and allocates weight to each class depending upon their frequency and after each iteration distant samples of class label created based on their weight. The process is repeated till all desired samples of the desired size are created. The allocated weight is proportionate to the class label's frequency in the current sample set. The higher the frequency, the lesser the allocated weight[34].

Hoeffding Tree

Hoeffding tree (HT) is a DT which is used for the classification of the class label and incremental learning and comprises 3 nodes: root, test, and leaf node. HT is famous for its efficiency and it generates similar trees as the first tree generated from the first training sets. HT learns from a large scale of networking data from which a small amount of data is sufficient to choose the best splitting attribute. The major disadvantage of HT is that when a tie occurs then it is unable to classify the data into trees[35].

Decision Table

A decision table (DT) is a hierarchy structure in which the values of pairs of attributes are used to break down each entry in a high level table into two tables. Dimension stacking is comparable to the structure of the decision table. A visualization technique is described by a decision tree that makes it possible for people with no prior knowledge of machine learning to comprehend a model based on a number of attributes. Numerous forms of interaction are used to make this representation more useful than earlier static versions. The DT is a simple visualization for specifying what to do in particular situations. Based on the specified criteria, this algorithm generates a series of actions as an output. In computer programming, the information in decision tables could be represented as decision trees or a sequence of if-then-else and switch-case instructions. In general, it is a tabular structure with rows and columns that illustrates decision rule[36].

K-Nearest Neighbor

The KNN algorithm is an easy and uncomplicated ML algorithm that uses a nonparametric approach for the classification which has a variety of applications like intrusion detection, data mining, speech recognition, text classification, and a variety of other topics. KNN is used for both regression and classification issues. It is, however, the best option for classification. It's a sluggish learner who just saves all of the training information. The purpose of this information is to look for patterns in both new and old data. It takes a lengthy time to compute the Euclidean distance, which is used to allocate the test data to the KNN class [37].

Decision Stump

An ML-based one level decision tree is referred to as a decision stump. A single root node that connects to every other node creates DT. Based just on a single input value, a DS provides a prediction. Several variations are possible based on the type of input feature. One split decision tree is produced using the DS operator. The created tree can be used to classify previously unidentified circumstances, but when paired with other operators, such the AdaBoost classifier, it is far more powerful. This method exemplifies the principles of machine learning and is a fantastic illustration of best practices. The DS is a binary classification algorithm. Binary classification is a method for classifying two categories, such as 0 and 1. The decision stump's idea is straightforward. Pick a point that can best separate data and just focus on one feature at a time. A one level decision tree (DT) classifier is known as a decision stump since it only considers one input value when making choices[38].

Naïve Bayes

The Naive Bayes (NB) algorithm, which is used for classification, is based on the Bayes theorem. Each pair of features in the NB set of algorithms is characterized as being independent of the others. Despite recent advances, machine learning has shown to be simple, rapid, and precise. It may be used to many different jobs, although it is particularly good at NIDS and NLP problems. The Naive Bayes method is used in many classification tasks. When some prior traffic data is available, the Naive Bayes approach predicts the traffic class using conditional probability.

The conditional probability and the class probability are the two probabilities that are part of Naive Bayes. The probability of each class is calculated by dividing the frequency of occurrence of each class by the total number of cases. Other classifiers are slower than Naive Bayes [39].

Boosting and Bagging

Boosting is a technique that combines predictions that belong to distinct types while bagging is the simplest approach to combine projections that belong to the same type. A uniform weak learner model is used by boosting, and it has several functions. AdaBoost classifier is a boosting technique used in this work. This method uses gradual, adaptive information acquisition to improve learning algorithm predictions. Boosting and bagging try to reduce bias rather than variance. A homogeneous weak learner's model called bagging combines individual parallel learning in order to determine the model average[40].

Results and Discussion

In this section, the outcomes of the proposed RC-based model are presented, and the results of the other used methodologies are then compared in terms of six performance metrics such as: accuracy, FPR, TPR, precision, MCC, and ROC area.

Table 4. Random Committee Based Model Results

Metrics	NSL-KDD	Kaggle
Accuracy	99.9103	99.8889
Precision	0.999	0.999
TPR	0.999	0.999
FPR	0.001	0.001
MCC	0.998	0.998
ROC Area	1	1

Table 4 shows that the RC based model gives better results than the previous literature presented in this study. RC based model gives the same precision, TPR, FPR, MCC and ROC area on both datasets while in terms of accuracy RC based model gives better results on the NSL-KDD dataset than the Kaggle dataset.

Table 5. Kaggle Dataset Results

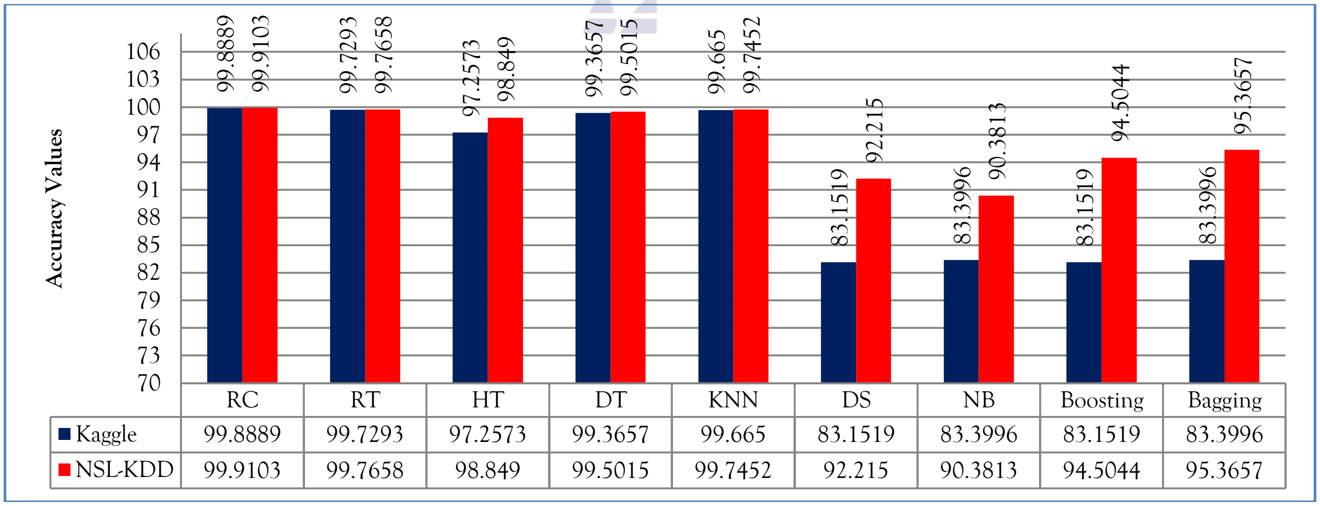
Applied Techniques	Accuracy	Precision	TPR	FPR	MCC	ROC Area
Random Tree	99.7293	0.997	0.997	0.002	0.996	0.998
Hoeffding Tree	97.2573	0.971	0.973	0.018	0.954	0.989
Decision Table	99.3657	0.994	0.994	0.006	0.989	0.997
KNN	99.665	0.997	0.997	0.002	0.994	0.997
Decision Stump	83.1519	0.97	0.832	0.12	0.943	0.882
Naïve Bayes	83.3996	0.91	0.834	0.046	0.786	0.966
Boosting	83.1519	0.925	0.832	0.12	0.844	0.952
Bagging	83.3996	0.91	0.834	0.046	0.786	0.92

Table 5 shows the results of all techniques such as a random tree, hoeffding tree, decision table, KNN, decision stump and NB applied on the Kaggle dataset using 10 FCV in terms of accuracy, precision, TPR, FPR, MCC and ROC area. It is clear from the Table 5 that the random tree performs very well among these techniques but from Table1 it concludes that the proposed model performs much better than all these techniques in terms of all performance metrics too.

Table 6. NSL-KDD Dataset Results

Applied Techniques	Accuracy	Precision	TPR	FPR	MCC	ROC Area
Random Tree	99.7658	0.998	0.998	0.002	0.995	0.998
Hoeffding Tree	98.849	0.989	0.988	0.012	0.977	0.995
Decision Table	99.5015	0.995	0.995	0.005	0.99	0.999
KNN	99.7452	0.997	0.997	0.003	0.995	0.998
Decision Stump	92.215	0.922	0.922	0.079	0.844	0.92
Naïve Bayes	90.3813	0.905	0.904	0.101	0.807	0.966
Boosting	94.5044	0.945	0.945	0.057	0.89	0.988
Bagging	95.3657	0.954	0.954	0.048	0.907	0.988

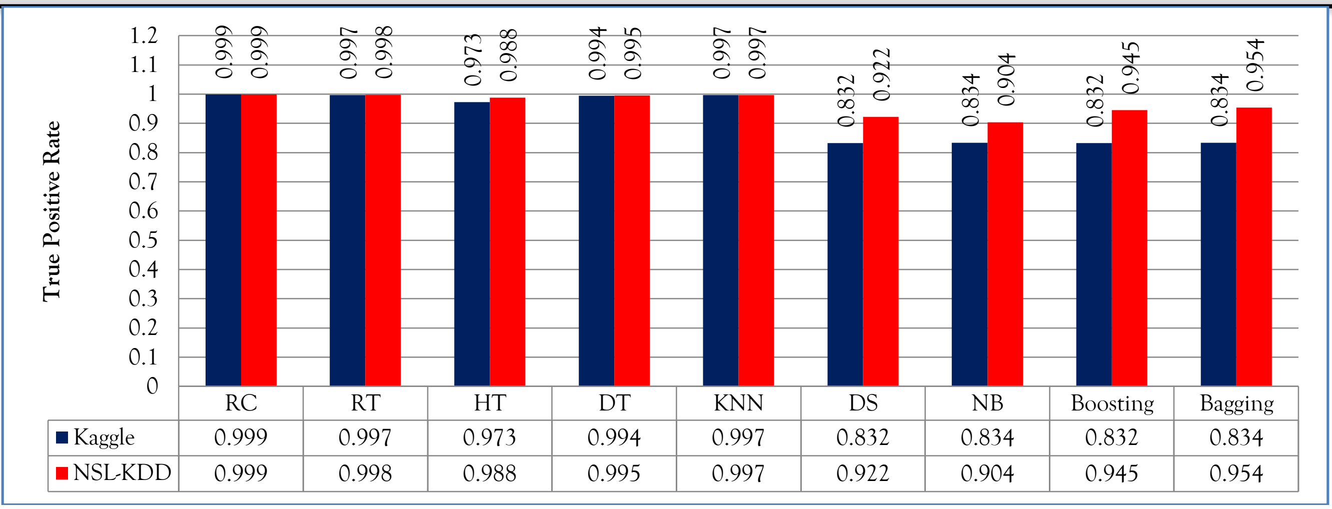
Table 6 shows the results of all applied techniques such as a random tree, hoeffding tree, decision table, KNN, decision stump and NB applied on the NSL-KDD dataset using 10 FCV in terms of accuracy, precision, TPR, FPR, MCC and ROC area. It is clear from Table5 and 6 that the random tree performs very well among these algorithms but Table 1 concludes that the RC-based model performs well among all algorithms too in terms of all performance metrics.



	RC	RT	HT	DT	KNN	DS	NB	Boosting	Bagging
■ Kaggle	99.8889	99.7293	97.2573	99.3657	99.665	83.1519	83.3996	83.1519	83.3996
■ NSL-KDD	99.9103	99.7658	98.849	99.5015	99.7452	92.215	90.3813	94.5044	95.3657

Figure 4. Accuracy Metrics of the Proposed Methods Evaluated on Both Datasets

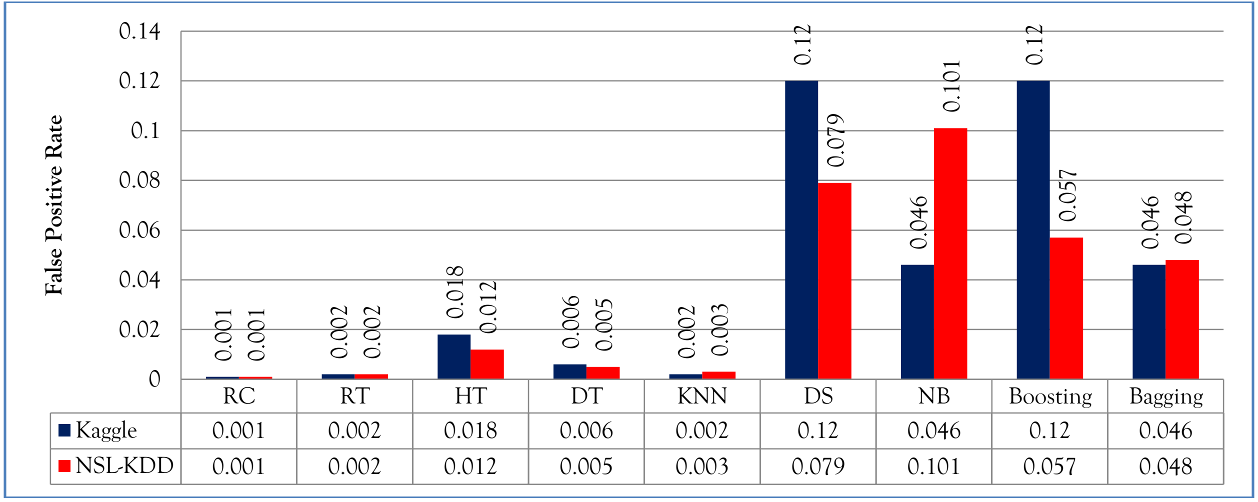
Fig. 4 shows the comparison of random tree, hoeffding tree, decision table, KNN, decision stump, NB and proposed RC based algorithm with having 99.9103 accuracies for NSL-KDD dataset and 99.8889 accuracies for Kaggle dataset which are highest among all as compared to the other applied algorithms.



	RC	RT	HT	DT	KNN	DS	NB	Boosting	Bagging
🟦 Kaggle	0.999	0.997	0.973	0.994	0.997	0.832	0.834	0.832	0.834
🟥 NSL-KDD	0.999	0.998	0.988	0.995	0.997	0.922	0.904	0.945	0.954

Figure 5. True Positive Rate Metrics of the Proposed Methods Evaluated on Both Datasets

Fig. 5 shows the comparison of random tree, hoeffding tree, decision table, KNN, decision stump, NB and proposed RC based algorithm with having 0.999 true positive rates for both NSL-KDD and Kaggle datasets which is highest among all as compared to the other applied algorithms.



	RC	RT	HT	DT	KNN	DS	NB	Boosting	Bagging
🟦 Kaggle	0.001	0.002	0.018	0.006	0.002	0.12	0.046	0.12	0.046
🟥 NSL-KDD	0.001	0.002	0.012	0.005	0.003	0.079	0.101	0.057	0.048

Figure 6. False Positive Rate Metrics of the Proposed Methods Evaluated on Both Datasets

Fig. 6 shows the comparison of RT, hoeffding tree, decision table, KNN, decision stump, NB and proposed RC based algorithm with having 0.001 false-positive rates for both NSL-KDD and Kaggle datasets which are lowest among all as compared to the other applied algorithms.

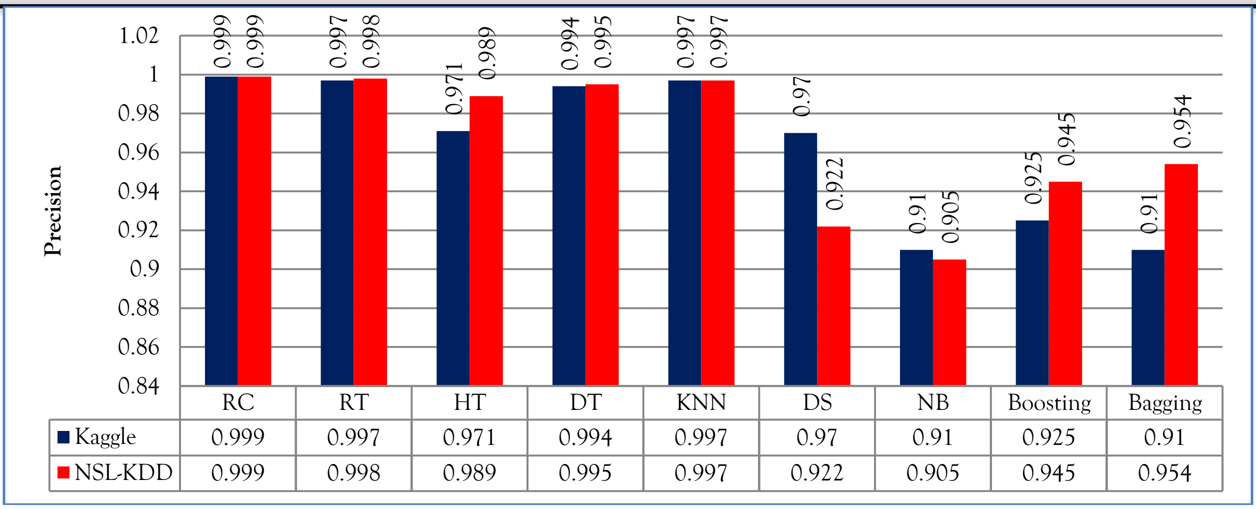


Figure 7. Precision Metrics of the Proposed Methods Evaluated on Both Datasets

Fig. 7 shows the comparison of RT, hoeffding tree, decision table, KNN, decision stump, NB and proposed RC based algorithm with having 0.999 precision for both NSL-KDD and Kaggle datasets which is lowest among all as compared to the other applied algorithms.

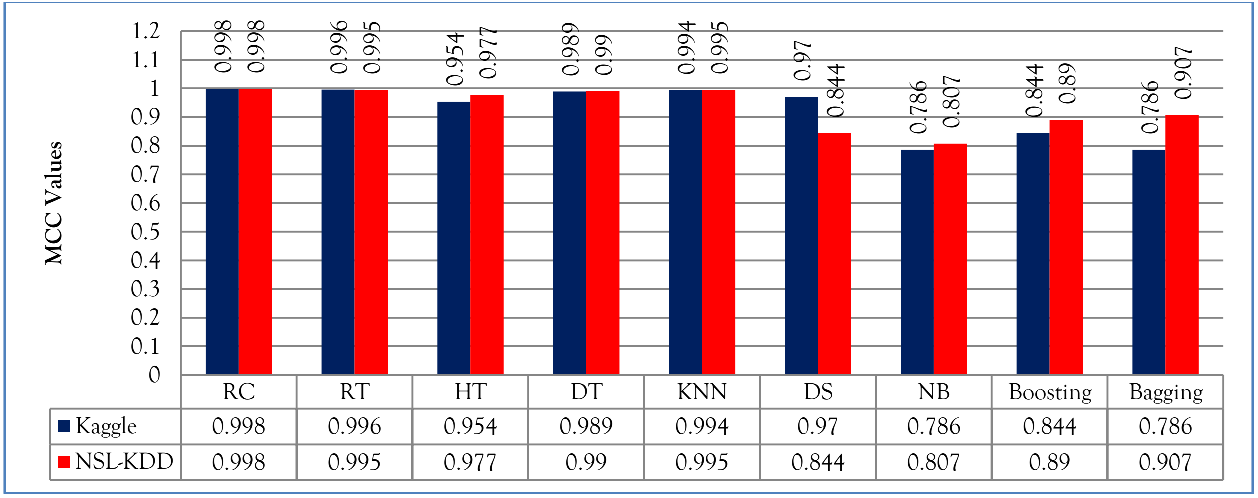
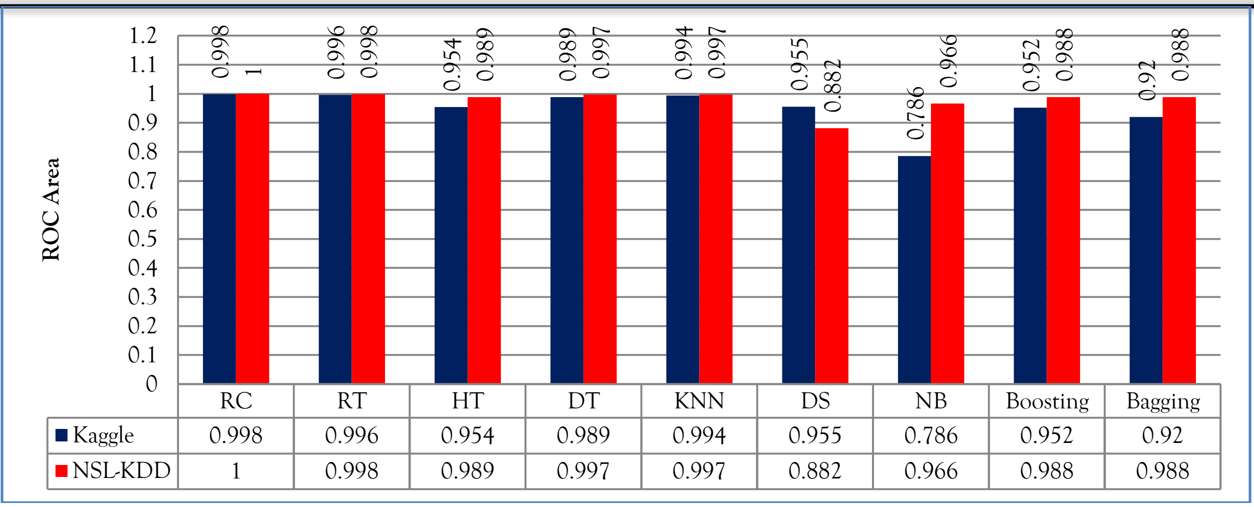


Figure 8. Mathews Correlation Coefficient Metrics of the Proposed Methods Evaluated on Both Datasets

Fig. 8 shows the comparison of RT, hoeffding tree, decision table, KNN, decision stump, NB and proposed RC based algorithm with having 0.998 MCC for both NSL-KDD and Kaggle datasets that are the best among all as compared to the other applied algorithms.



	RC	RT	HT	DT	KNN	DS	NB	Boosting	Bagging
■ Kaggle	0.998	0.996	0.954	0.989	0.994	0.955	0.786	0.952	0.92
🟥 NSL-KDD	1	0.998	0.989	0.997	0.997	0.882	0.966	0.988	0.988

Figure 9. ROC Area Metrics of the Proposed Methods Evaluated on Both Datasets

Fig. 9 shows the comparison of RT, hoeffding tree, decision table, KNN, decision stump, NB and proposed RC based algorithm with having 1 ROC area for both NSL-KDD and Kaggle datasets which is the highest and most accurate among all as compared to the other applied algorithms.

Discussion

Based on experimental analysis, the Random Committee-based model of intrusion detection has shown itself to be superior to a variety of well-known machine-learning methods when used with both NSLKDD and Kaggle. In the NSLKDD dataset, Random Committee model recorded highest accuracy- 99.9103%. The metrics of precision, true-positive rate, Matthews Correlation Coefficient and false-positive rate are respectively: (0.999), (0.999), (0.998), and (0.001). The ROC curve has an area of 1.0, which indicates that it is a highly discriminative level between a normal and a malicious network activity. Random Tree provided the second best results of individual algorithms with 99.7658 of accuracy on NSLKDD and 99.7293 of accuracy on Kaggle. The next was KNN with 99.7452 percent on NSLKDD and 99.665 percent on Kaggle. Decision Table was also considered to give reasonably working results with greater than 99 percent accuracies on both the data sets, but only the ensemble techniques including Boosting and Bagging gave intermediate effectiveness. Hoeffding Tree delivered the least accurate results, being 98.849 and 97.2573 on NSLKDD and Kaggle, respectively; Decision Stump and Naive Bayes also performed worse with a below-93% accuracy level and a significantly high-false-positive rate.

The results indicate that the majority of algorithms were more successful on NSL-KDD than on Kaggle, and, therefore, the patterns in the former data were more discrete. However, Random Committee demonstrated high accuracy, low false-positive rates, which prove its soundness and Applicability of results in the environment of both datasets. The Random Committee model is an ideal network-intrusion detection approach. Empirical studies show that it has low rates of false positives, high rates of detection, and has a tremendous record of stopping false alarms. This kind of superiority in ensemble learning settings brings out the practicality of the ensemble methods in network-intrusion detections. Random Committee prevents excessive levels of bias and variance, because in contrast to single base classifiers, several classifiers are combined by aggregation, thus allowing a stronger, less biased and more accurate prediction. The high level of performance over different data sets makes the model an effective and powerful instrument in addressing modern cyber security issues and forms the basis of developing more advanced intrusion-detection systems.

Conclusion

In this paper, a ML based model is designed and tested for NIDS, using multiple performance metrics: including accuracy precision, TPR and FPR etc are also considered for model efficiency. RC-based model for NIDS is presented and the results are compared to other ML techniques: RT, HT, DT, KNN, DS and NB using two datasets: NSL-KDD and Kaggle for classification. The datasets are classified using 10 FCV and all techniques results are presented in terms of performance metrics. However, in the case of both datasets, RC provides the best results for the classification of normal and anomaly classes. The proposed RC model can be beneficial for researchers to construct an improved NIDS. The proposed RC model achieves the highest accuracy of 99.9%. It is concluded from Table 4, 5 and 6 that the proposed RC based model performed the best among all other techniques and also from the techniques mentioned in literature review section. In the future, the most advanced techniques like graph neural networks or deep learning must be applied on real-time datasets. Furthermore, for real time analysis IoT based systems must be applied to multiclass datasets.

REFRENCES

[1] Diana, L., Dini, P., & Paolini, D., “Overview on intrusion detection systems for computers networking security,” Computers, 14(3), 8, 2025.

[2] M. Sarnovsky and J. Paralic, “Hierarchical intrusion detection using machine learning and knowledge model,” Symmetry (Basel)., vol. 12, no. 2, pp. 1–14, 2020.

[3] Awad, Z., Zakaria, M., & Hassan, R. “An enhanced ensemble defense framework for boosting adversarial robustness of intrusion detection systems,” Scientific Reports, 15(1), 14177, 2025.

[4] S. A. Hussein, A. A. Mahmood and E. O. Oraby, “Network intrusion detection system using ensemble learning approaches,” Webology, vol. 18, no. Special Issue, pp. 962–974, 2021.

[5] S. Razdan, H. Gupta and A. Seth, “Performance analysis of network intrusion detection systems using j48 and naive bayes algorithms,” 2021 6th Int. Conf. Converg. Technol. I2CT 2021, pp. 1–7, 2021.

[6] Anandaram, H., Mishra, N. K., & Nidhya, M. S., “Evaluation of Artificial Intelligence Techniques in Disease Diagnosis and Prediction. In Handbook of Artificial Intelligence and Wearables” (pp. 124-144). CRC Press, 2024.

[7] M. Data and M. Aritsugi, “T-DFNN: an incremental learning algorithm for intrusion detection systems,” IEEE Access, vol. 9, pp. 154156–154171, 2021.

[8] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu et al., “Hierarchical adversarial attacks against graph neural network based IoT network intrusion detection system,” IEEE Internet of Things Journal, vol. 9, no. 12, June 15, 2022.

[9] D. Chou and M. Jiang, “A Survey on Data-driven Network Intrusion Detection,” ACM Computing Survey, vol. 54, no. 9, pp. 1–36, 2022.

[10] S. Lee, A. Abdullah, N. Jhanjhi and S. Kok, “Classification of botnet attacks in IoT smart factory using honeypot combined with machine learning,” PeerJ Computer Science, vol. 7, pp. 1–23, 2021.

[11] “Global ransomware damage costs to exceed $265 billion by 2031 - EIN Presswire,”(accessed Jun. 03, 2025) https://www.einnews.com/pr_news/542950077/global-ransomware-damage-costs-to-exceed-265-billion-by-2031.

[12] “Cybercrime to cost the world $10.5 trillion annually by 2025.” (accessed Jun. 17, 2025) https://cyber-securityventures.com/cybercrime-damages-6-trillion-by-2024/.

[13] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa and C. F. M. Foozy, “Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset,” IEEE Access, vol. 9, pp. 22351–22370, 2021.

[14] P. Dini and S. Saponara, “Analysis, design, and comparison of machine-learning techniques for networking intrusion detection,” Designs, vol. 5, no. 1, pp. 1–22, 2021.

[15] T. Kim and W. Pak, “Hybrid classification for high-speed and high-accuracy network intrusion detection system,” IEEE Access, vol. 9, pp. 83806–83817, 2021.

[16] H. Alqahtani, I. H. Sarker, A. Kalim, S. M. Minhaz Hossain, S. Ikhlaq et al., “Cyber

intrusion detection using machine learning classification techniques, " vol. 1235 CCIS. Springer Singapore, 2020.

[17] I. Ullah and Q. H. Mahmoud, "Design and development of a deep learning-based model for anomaly detection in IoT networks," IEEE Access, vol. 9, pp. 103906–103926, 2021.

[18] M. Sarhan, S. Layeghy and M. Portmann, "Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection," pp. 1–12, 2021, [Online]. Available: http://arxiv.org/abs/2104.07183.

[19] J. Liu, B. Kantarci and C. Adams, "Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset," WiseML 2020 - Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, pp. 25–30, 2020.

[20] I. H. Sarker, Y. B. Abushark, F. Alsolami and A. I. Khan, "IntruDTree: A machine learning based cyber security intrusion detection model," Symmetry (Basel)., vol. 12, no. 5, pp. 1–15, 2020.

[21] S. Mane and D. Rao, "Explaining network intrusion detection system using explainable AI framework," Cryptography and Security, vol. 1, no. Ml, pp. 1–10, 2021.

[22] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey and R. T. Goswami, "An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset," Cluster Computing, vol. 23, no. 2, pp. 1397–1418, 2020.

[23] L. Le Jeune, T. Goedeme and N. Mentens, "Machine learning for misuse-based network intrusion detection: overview, unified evaluation and feature choice comparison framework," IEEE Access, vol. 9, pp. 63995–64015, 2021.

[24] K. A. Taher, B. M. Y. Jisan and M. M. Rahman, "Network intrusion detection using supervised machine learning technique with feature selection," 1st International Conference on Robotics, Electrical and Signal Processing Techniques, ICREST 2019, pp. 643–646, 2019.

[25] Dharini, N., Katiravan, J., & Shakthi, S. P., "Botnet attack detection in iot devices using ensemble classifiers with reduced feature space." International Research Journal of Multidisciplinary Technovation, 6(3), 274-295, 2024.

[26] López-Vizcaíno, M., Nóvoa, F. J., Fernández, D., & Cacheda, F., "Time Aware F-Score for Cybersecurity Early Detection Evaluation," Applied Sciences, 14(2), 574, 2024.

[27] D. S. Berman, A. L. Buczak, J. S. Chavis and C. L. Corbett, "A survey of deep learning methods for cyber security," Information, vol. 10, no. 4, 2019.

[28] C. M. Hsu, M. Z. Azhari, H. Y. Hsieh, S. W. Prakosa and J. S. Leu, "Robust network intrusion detection scheme using long-short term memory based convolutional neural networks," Mobile Networks and Applications, vol. 26, no. 3, pp. 1137–1144, 2021.

[29] Wang, X., Qiao, Y., Xiong, J., Zhao, Z., Zhang, N., Feng, M., & Jiang, C., "Advanced network intrusion detection with tabtransformer" Journal of Theory and Practice of Engineering Science, 4(03), 191-198, 2024.

[30] Y. Pacheco and W. Sun, "Adversarial machine learning: a comparative study on contemporary intrusion detection datasets," ICISSP 2021 - Proceedings of the 7th International Conference on Information Systems Security and Privacy, no. Icissp, pp. 160–171, 2021.

[31] N. Mary, B. Khan, A. A. Asiri, F. Muhammad, S. Khan et al., "Heart disease risk prediction expending of classification algorithms," Computers, Materials and Continua, vol. 73, no. 3, pp. 6595-6616, 2022.

[32] Z. A. Baig, S. Sanguanpong, S. N. Firdous, V. N. Vo, T. G. Nguyen et al., "Averaged dependence estimators for DoS attack detection in IoT networks," Future Generation Computer Systems, vol. 102, pp.

198–209, 2020.

[33] W. W. Lo, S. Layeghyy, M. Sarhanz, M. Gallagher and M. Portmann, "E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT," IEEE/IFIP Network Operations and Management Symposium, pp. 1-9, April, 2022.

[34] Amru, M., Kannan, R. J., Ganesh, E. N., Muthumarilakshmi, S., Padmanaban, K., Jeyapriya, J., & Murugan, S., "Network intrusion detection system by applying ensemble model for smart home," International Journal of Electrical and Computer Engineering, 14(3), 3485-3494, 2024.

[35] Data, M., & Aritsugi, M., "AB-HT: An ensemble incremental learning algorithm for network intrusion detection systems," International Conference on Data Science and Its Applications (ICoDSA) (pp. 47-52). IEEE, 2022.

[36] Azam, Z., Islam, M. M., & Huda, M. N., "Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision tree," IEEE Access, 11, 80348-80391, 2023.

[37] Mohy-Eddine, M., Guezzaz, A., Benkirane, S., & Azrour, M., "An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection," Multimedia Tools and Applications, 82(15), 23615-23633, 2023.

[38] Liu, Q., Hagenmeyer, V., & Keller, H. B., "A review of rule learning-based intrusion detection systems and their prospects in smart grids," IEEE Access, 9, 57542-57564, 2021.

[39] Jeevaraj, D., "Feature selection model using naive bayes ML algorithm for WSN intrusion detection system," International journal of electrical and computer engineering systems, 14(2), 179-185, 2023.

[40] Zhang, Z., Kong, S., Xiao, T., & Yang, A., "A Network Intrusion Detection Method Based on Bagging Ensemble," Symmetry, 16(7), 850, 2024.