Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 6, 2026

COMPARATIVE ANALYSIS OF MACHINE LEARNING TECHNIQUES FOR NETWORK INTRUSION DETECTION IN CYBER SECURITY WITH A DIVERSE METRIC-BASED PERFORMANCE ASSESSMENT

[1]Farhan Tariq, [2]Hina Kanwal, [3]Shaheena Azam, [4]Jowaria Shereen, [5]Shakeela Maqsood

[1]Department: School of Computing, Engineering and Information Sciences, Northumbria University, London, UK
[2]Department: Computer Science, Comsats University Islamabad
[3]Department: Computer Science, Comsats University, Islamabad, Pakistan
[4]Department: Software Engineering, Bahria University, Islamabad, Pakistan
[5]Department Computer Science, The Islamia University Bahwalpur
[1]Farhantariq5251@gmail.com, [2]kanwalhina636@gmail.com, [3]shaheena21ms@gmail.com, [4]jawariafaisal123@gmail.com, [5]shakeelamaqsood01@gmail.com

DOI:-https://doi.org/10.5281/zenodo.20570609

Keywords
Cyber security; Machine Learning; Intrusion Detection; Pareto Principle; Cross Validation

Article History
Received: 13 May, 2026
Accepted: 05 June, 2026
Published: 06 June, 2026

Corresponding Author: *

Abstract

In modern communication and networking, the safe and reliable transfer of data is a necessity of time because the number of intruder attacks on computer networks aims to gain access to crucial information. To protect the network data from any malicious attack, the network intrusion detection systems (NIDSs) play the most critical role. It analyzes the data pattern and secures the network from any attack. This pattern analysis is not possible manually due to the large scale of data; however, machine learning (ML) is a powerful technique to analyze the large scale of data patterns and detect any malicious threats. In this work, we integrated ML with NIDS to analyze and monitor the networking data. We have applied six supervised ML techniques, which include Random, Hoeffding, and Decision Tree, Averaged One-Dependence Estimators, Instance-based KNN, and Naive Bayes, during the experiment and also considered six performance assessment criteria, which include accuracy, precision, true and false positive rates, Matthew correlation coefficient, and receiver operating characteristic area for the three different datasets. The Pareto principle is considered for the training and testing data. According to the results, A1DE is the best model among the applied techniques; it identifies patterns in the data with 99.9964% accuracy, which establishes a foundation for further research. The researchers use these findings as a starting point for determining which cyber-related attributes should be prioritized to create the most effective and successful NIDS.

1. Introduction

This Information and communication technology (ICT) use is on the rise these days, and with it come cyber attacks on these systems. The research community carried out a number of investigations to provide a safe detection method to fight anomalous threats. To guarantee secure communication, network security researchers and experts are particularly focused on detecting and preventing cyber attacks [1]. Many businesses and organizations invest a significant amount of their budgets on network security in order to protect the Confidentiality, Integrity, and Availability (CIA) of the data [2]. It can be difficult to distinguish useful information from vast volumes of valuable data in a digitally linked society, which makes it more difficult to identify malicious or intruding data. Network Intrusion Detection Systems (NIDS) are essential for network protection and risk reduction in such a demanding environment [3]. Machine learning (ML) models are employed in network security since traditional security measures like firewalls and antivirus software cannot identify and stop new threats [4]. Intrusion is the term for unauthorized access to a network with malevolent intent. Network intrusion detection systems (NIDS) are hardware and software systems that continually monitor all network activities. NIDS work incredibly well, despite having significant issues including poor accuracy and a high false alert rate [5].

In order to guarantee safe and secure communication, researchers primarily focus on network security and use firewalls, antivirus software, and NIDS. The best choice in this field is to use deep learning (DL) and machine learning (ML). Machine learning (ML) and deep learning (DL) are examples of artificial algorithms (AI) that are being used realistically to predict both normal and abnormal behaviors [6]. ML-based models make it possible to extract valuable information from incredibly complex data. ML-based NIDS improves accuracy and requires less human expertise because it learns from the data pattern itself [7]. Finding, evaluating, and stopping any cyberattack on a network's host computer are the main objectives of NIDS. Anomaly-based, signature-based, or a mix of the two are commonly used by NIDS for effective detection [34]. NIDS collects data from individual and large numbers of network-connected machines to detect abuse that is taking on both inside and outside of a network [8]. The NIDS taxonomy is displayed in Fig. 1 [9].

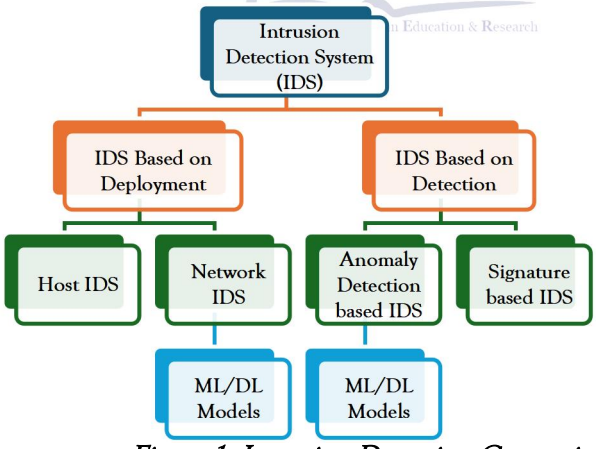


Figure 1: Intrusion Detection Categorization

ICT is also utilized in smart businesses, where data may easily flow throughout all processes and all gadgets are wirelessly connected. These industries require the most effective, organized, and integrated security approaches [10].

Researchers came up with a number of NIDS models, each with pros and cons, while keeping all of these concerns in mind. This work's primary contributions are as follows: first, it suggests many machine learning models for NIDS based on binary class and multiclass datasets, including NSL-KDD, UNSW, and Kaggle, respectively. Second, there are a lot of characteristics in all three datasets, which causes overfitting. The suggested model considerably lessens the overfitting problem in order to solve this problem. Thirdly, whereas 10 fold cross validation was employed in earlier studies, we have applied the Pareto principle, often known as the 80/20 rule, to every dataset in our work.

Finally, all datasets have labeled data, and we employed supervised machine learning approaches to create the best model for NIDS based on labeled data analysis [34]. The next sections explore the comparison between the suggested model and popular machine learning models.

The remainder of the research paper is structured as follows: the literature review is presented in Section 2, the suggested A1DE model and alternative ML models are presented in Section 3, and the experimental model is thoroughly described in Section 4. The findings are displayed in Section 5, and the paper's conclusion is given in Section 6.

2. Literature Review

Machine learning techniques for NIDS have become a major research topic in recent years. Among the several techniques employed is anomaly and signature-based NIDS, or a combination of the two. Numerous research articles about signature-based NIDS may be found in the literature.

Using a CICIDS2017 dataset, the authors assess the efficacy of 48 ML-based anomaly detection systems (AIDS). There is discussion of both supervised and unsupervised paradigms; also, the former has a detection rate of 99.28%, while the latter has a detection rate of 60.06 percent. The findings encourage meticulous feature selection and recommend building deep learning models for in-depth analysis. The results demonstrate that the usage of K-mean yields the least optimum output, whereas the artificial-neural-network (ANN) design produces the best results. Lastly, the study notes that integrating the feature-selection strategy with machine learning-based AIDS can result in a notable increase in AIDS accuracy [11].

The accuracy and precision of the two ML algorithms—artificial neural networks (ANN) and k-nearest neighbors (KNN)—were compared with those of NIDS in this study. In contrast to the ANN model, which achieved 0.9923, 0.9910, and 0.9926, 0.9920, the KNN model achieved 0.9957 accuracy, 0.9949 precision, 0.9959 TPR, and 0.9956 TNR [35]. Both techniques operate differently on the databases that are accessible, and the primary restrictions are the size of the feature selection and the time density computations. KNN therefore operates on certain data [12].

The authors used ISCXIDS2012 and CICIDS2017 to investigate intrusion detection. A hybrid system that combines a packed and session classifier achieves 97.37 percent accuracy on ISCXIDS2012 and 99.8 percent accuracy on CICIDS2017. Adaboosted decision trees, random forests, deep neural networks, SMOTE+RF, support vector machines, TSE, including rotating forests, extreme learning machines, and gradient boosting trees are all compared to this model [36]. Despite being quite precise, the method's main disadvantages are its high cost and complexity [13].

The authors examined data-driven IDS created using a 10-fold cross validation (FCV) technique using the KDDcup99 data set. They used accuracy, precision, and recall as performance metrics after training two ML models, Random Forest (RF) and Decision Tree (DT). The accuracy, precision, and recall scores of the RF were 94% against 93% of the DT, 93% against 92%, and 94% against 92% of the DT [14].

The Convolutional neural networks (CNNs) in 1D, 2D, and 3D are developed by authors to identify network abnormalities. The Pareto rule is used to divide the dataset, according to which 80% of it should be utilized for training and 20% for testing. All CNN-based models yield high accuracy for four classes. The detection rates for Normal, Scan, Theft, and DoS were 99.90%, 99.91%, 98.10%, and 99.96%, respectively. The FNR was 0.67% while the FPR was 0.05%. According to [15], the minimum recognition rates for the CNN1D, CNN2D, and CNN3D models were 99.74%, 99.42%, and 99.03% respective [37].

For data categorization, the authors suggest two machine learning algorithms: Random Forest (RF) and Deep Feed Forward (DFF) classifiers. A 70:30 split between training and testing sets is applied to each dataset. By raising the data rate and decreasing the false alarm rate, both classifiers are able to attain high detection accuracy. The accuracy of the DFF classifier for NF-CSE-CIC-IDS2018-v2 is 99.24%. For NF-CSE-CIC-IDS2018-v2, the RF classifier achieves an accuracy of 99.47% [16].

The authors of this work create a decision-making model that resembles a tree and choose particular features according to their scores and ranks [38]. The Python programming language is used to partition the data into discrete subsets for

simple feature encoding and scaling, reducing model variance and over-fitting problems. In addition to increasing the prediction rate for unknown threats, the suggested approach decreases complexity. With 0.98 precision, 0.981 recall, 0.98 F-score, and 0.981 accuracy, the suggested IntrDTree model is ultimately shown to be more effective than the conventional models [17].

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 [39]. 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. [18].

This study compares the SVM model with an ANN-based model that uses wrapper feature selection. Trial-and-error techniques are used for classification on the NSL-KDD dataset. With a detection rate of 94.02%, the findings demonstrate that the ANN with wrapper feature selection works well. The accuracy of the ANN model is 94.02% for 17 features and 83.68% for 35 features, whereas the accuracy of the SVM model is 81.78% for 17 features and 82.34% for 35 features [19].

Previous studies had two main drawbacks. First of all, the authors only employed accuracy as a performance criterion to assess the model's effectiveness on a single dataset. Secondly, they did not employ any novel machine learning techniques. The majority of earlier research ignored multiclass real-time data sets in favor of binary categorized datasets. To get around these issues, this study employed a novel supervised machine learning approach called random committee (RC). It then utilized a variety of performance indicators to evaluate the model's effectiveness on two distinct datasets: binary classified and multiclass.

3. Proposed Machine Learning Model

This section discusses the proposed model, A1DE for NIDS.

Averaged One Dependence Estimators

Averaged One Dependence Estimators (A1DE) is a probabilistic classification algorithm on which intrusion and anomaly detection have been extensively studied because of it accuracy and efficiency [20]. As opposed to the classical Naive Bayes classifier, which uses conditionality of all features, given the class label, A1DE loosens this assumption, that is, each feature is allowed to be dependent, not only on the class but also on a single feature. This enables it to be more effective in dealing with correlated attributes that are prevalent in the network traffic data like packet size, duration, and protocol type. A1DE averages the predictions of many one dependence classifiers to minimize bias and variance to achieve greater classification errors and strength. A1DE has shown high detection rates of common as well as rare attacks, including User to Root (U2R) and Remote to Local (R2L) attacks. It is well adapted to real-time applications, since it can train and classify very fast, and tends to be much better at it than Naive Bayes, aside from being more computationally efficient than any complex model (e.g. deep neural networks or ensemble methods) [21]. Due to these reasons, A1DE can be regarded as a good method to create a reliable and scalable intrusion and anomaly detection systems.

4. Experimental Setup

In order to demonstrate ML classification techniques for IDS, this study analyzes the performance of two distinct datasets that were downloaded from the Kaggle repository. The whole study procedure is depicted in Fig. 2. Each dataset goes through a preprocessing step after being chosen, with the twin goals of converting the class attribute from a numerical to a categorical value and recovering missing values. After all, the outcomes of applied machine learning techniques to every dataset are assessed using various metrics to show how well a particular technique performs.

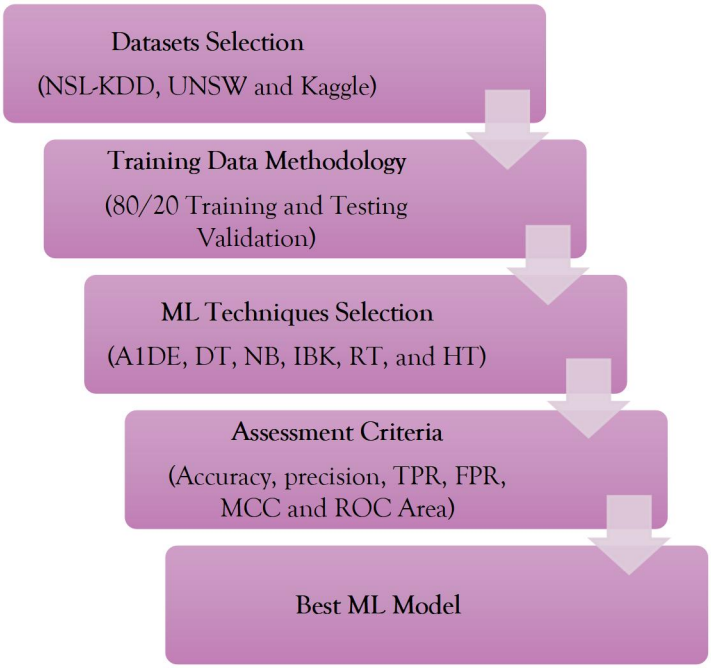


Figure 2. Flow Diagram of this Work

Furthermore in this section, three datasets: NSL-KDD UNSW, and Kaggle, 80/20 rule or pareto principle, multiple performance metrics with mathematical forms and applied techniques such as: A1DE, DT, NB, IBK, RT, and HT are presented.

Datasets

Both binary class datasets and multiclass datasets, such those from NSL-KDD and Kaggle, also a real-time dataset USSW are presented in this section. Cybersecurity-related research makes use of these databases. A known class label and a few characteristics make up each dataset. Each dataset contains numerical data, even when the total number of characteristics and occurrences varies.

Kaggle Dataset

There are five categories of samples in the Kaggle dataset: normal, DoS, r21, probe, and u2r. Table 1 displays the two sets of 125973 instances for 80% training and 20% testing in this adjusted dataset, which no longer includes any raw network data. Several characteristics and instance sets are included in this dataset, which includes both common attacks and some of the more prevalent ones from real-time networking. The most trustworthy dataset for evaluating IDS's real performance is this one [22].

UNSW Dataset

The raw network traffic of the UNSW-NB 15 dataset were created using the IXIA PerfectStorm software in UNSW Canberra's Cyber Range Lab in order to provide a combination of real modern everyday activities and simulated current attack behaviors. The tcpdump tool was used to gather 100 GB of the raw data (such as Pcap files). This dataset includes nine attack types: worms, reconnaissance, shellcode, analysis, backdoors, DoS, exploits, generic, and fuzzers [33]. A total of 49 characteristics with the class label are produced using twelve algorithms and the Argus and Bro-IDS tools [23].

NSL-KDD Dataset

The NSL-KDD dataset is frequently used as a reference for evaluating the efficacy of NIDS in research. This dataset includes several new assaults and is an upgraded version of KDD Cup 99. The main advantage of this collection is the removal of duplicate and superfluous entries. NSL-KDD is a binary classification with 41 characteristics that are classed as normal and abnormal. As shown in Table 1, this dataset's training and testing sets, which together include over 125,000 characteristics, are divided into two categories [24].

Table 1: *Datasets Statistics*

	Training	Testing
Kaggle	113376	12597
UNSW	175341	82332
NSL-KDD	113376	12597

10 Fold Cross Validation

Cross-validation is used to separate the dataset into training and testing sets. It is blended before being divided into K groups. While the testing model receives very little data, the training model receives a bulk data. Previously, researchers used the 10 FCV validation and in this work, pareto principle which means 80/20 rule is used. The procedure of cross-validation is shown in Fig. 3.

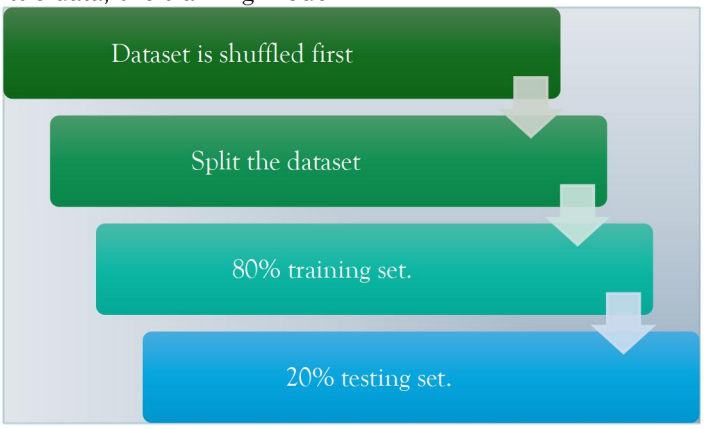


Figure 3. Pareto Principle Flowchart

Performance Assessments

Metrics that evaluate the effectiveness of a model using a representation often known as a confusion matrix are used to evaluate its efficacy. The confusion matrix determines how successful categorization models are. For every experiment in this study, binary categorization classes of positive (P) and negative (N) are used. Table 2 displays the four results that the confusion matrix produces based on the two classes [25].

Table 2: *Representation of Confusion Matrix*

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

The positive and negative class indicators are called True Positive (TP) and True Negative (TN), respectively. A false positive (FP) occurs when the model forecasts P but the real response is N. When the model predicts N but the right answer is P, this is known as a false negative (FN) [26].

The confusion matrix, a popular paradigm for performance evaluation, is used to compare the actual class values with the projected class values. The percentage of correctly classified samples to all examined cases is known as accuracy and the mathematical form is:

(TP+TN)/(TP+TN+FN+FP) (1)

By dividing an object's real positive values by its total positive values, one may calculate its precision and the mathematical form is:

TP/(TP+FP) (2)

The ratio of incorrectly identified negative samples to all negative samples is known as the false positive rate (FPR), whereas the ratio of all truly detected items to all genuine samples is known as the true positive rate (TPR). The classifier's trade-off between TP and FP is shown by the ordinate, TPR, and the abscissa, FPR, for each point. The mathematical forms of TPR and FPR are shown in the equation 3 and 4 respectively.

TP/(TP+FN) (3)

FP/(TN+FP) (4)

Although it is sensitive to the distribution of classes in a testing set, the Mathews Correlation Coefficient (MCC) is a balanced statistical score that accounts for both SN and SP. An accurate categorization is accomplished when the computed value is around 1 and the mathematical form is shown below [27].

(((TP*TN)-(FP*FN)))/√(((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))) (5)

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 which uses N randomly chosen variables at each node to fit several decision trees to a dataset are called Random Tree (RT). Each tree in these symmetrically distributed sets of random decision trees receives an equal sampling. By combining these evenly spaced trees, RT creates a model that is more realistic and precise. RT is a hybrid method that employs a single tree based on random forest (RF) principles, with k randomly selected characteristics at each node in the tree. As a result, random forests perform more accurately than a single tree. Following each iteration, distant samples of class labels are formed according on their weight. RT generates a sample of each class and assigns weight to each class based on their frequency. Until all the required samples of the required size are produced, the procedure is repeated. The weight allotted is based on how frequently the class label appears in the current sample. The weight allotted decreases with increasing frequency [28].

Hoeffding Tree

The Hoeffding tree, which consists of three nodes—the root, test, and leaf nodes is basically a decision tree and is used for class label categorization and incremental learning. The efficiency of HT is well known, and it produces trees that are comparable to the original tree created from the first training sets. The best splitting attribute may be selected using a little fraction of the vast quantity of networking data that HT learns from. The main drawback of HT is that it cannot categorize the data into trees when there is a tie [29].

K-Nearest Neighbor

The KNN method is a simple and straightforward machine learning technique that employs a nonparametric approach for classification. Among its many uses are intrusion detection, data mining, speech recognition, text classification, and many more. KNN is applied to problems involving both classification and regression. Nonetheless, it is the most effective categorization choice. This slow learner just stores all of the training data. Finding trends in both fresh and old data is the aim of this information. Calculating the Euclidean distance, which is required to assign the test data to the KNN class, is time-consuming [30].

Naïve Bayes

The The Naive Bayes (NB) is an approach to classification which is based on the Bayes theorem. Each pair of attributes in the NB set of algorithms is said to be independent of the others. Despite later advancements, machine learning has shown to be simple, fast, and accurate. Although it is highly effective at NIDS problems, it may be used for a variety of jobs. The Naive Bayes technique is used in a number of classification tasks. When some historical traffic data is available, this approach forecasts the traffic class using conditional probability. The conditional probability and the class probability are the two probabilities that comprise Naive Bayes. The probability of each class is calculated as the frequency of occurrence divided by the total number of cases [32]. Naive Bayes is faster than other classifiers for small datasets [31].

6. Results and Discussion

Six performance measures, including accuracy, precision FPR, TPR, MCC, and ROC area, and all results of the applied ML techniques such as A1DE, DT, NB, IBK, RT, and HT model are presented below. The results of all approaches are compared using

Table 3: *UNSW Dataset Results*

Applied Techniques	Accuracy	Precision	TPR	FPR	MCC	ROC Area
Averaged 1D Estimators	99.9964	0.994	0.994	0.005	0.988	1
Decision Tree	99.4243	1	1	1	1	1
Naïve Bayes	76.8243	0.802	0.768	0.205	0.572	0.864
Instance-Based KNN	98.721	0.987	0.987	0.013	0.974	0.987
Random Tree	99.2943	0.993	0.993	0.007	0.986	0.993

Hoeffding Tree	96.6028	0.966	0.966	0.038	0.932	0.981

In terms of accuracy, precision, TPR, FPR, MCC, and ROC area, Table 3 displays the outcomes of all methods, including an A1DE, DT, NB, IBK, RT, and HT model applied to the UNSW Dataset utilizing the Pareto principle. With the highest accuracy and lowest FPR among these methods, the A1DE model performs exceptionally well, as the table makes evident. Because the naïve bayes model cannot effectively handle large datasets, it is less accurate.

Table 4: *Kaggle Dataset Results*

Applied Techniques	Accuracy	Precision	TPR	FPR	MCC	ROC Area
Averaged 1D Estimators	99.5792	0.996	0.996	0.005	0.992	1
Decision Tree	99.5554	0.996	0.996	0.004	0.991	0.998
Naïve Bayes	98.9997	0.99	0.99	0.011	0.98	0.999
Instance-Based KNN	98.0668	0.981	0.981	0.021	0.961	0.989
Random Tree	92.2078	0.922	0.922	0.079	0.843	0.918
Hoeffding Tree	89.6634	0.897	0.897	0.106	0.792	0.965

Table 4 shows the results of all techniques, including an A1DE, DT, NB, IBK, RT, and HT model applied to the Kaggle Dataset using the Pareto principle, in terms of accuracy, precision, TPR, FPR, MCC, and ROC area. The table shows that the A1DE model performs very well, having the lowest FPR and the highest precision and accuracy among all these techniques. The Hoeffding Tree model is less accurate since it cannot handle multi-classed datasets well.

Table 5: *NSL-KDD Dataset Results*

Applied Techniques	Accuracy	Precision	TPR	FPR	MCC	ROC Area
Averaged 1D Estimators	99.7952	0.998	0.998	0.002	0.996	1
Decision Tree	99.7817	0.998	0.998	0.002	0.996	0.999
Naïve Bayes	99.7452	0.997	0.997	0.003	0.995	0.998
Instance-Based KNN	99.7658	0.998	0.998	0.002	0.995	0.998
Random Tree	98.849	0.989	0.988	0.012	0.977	0.995
Hoeffding Tree	90.3813	0.905	0.904	0.101	0.807	0.966

The accuracy, precision, TPR, FPR, MCC, and ROC area of all methods—including an A1DE, DT, NB, IBK, RT, and HT model—applied to the NSL-KDD Dataset employing the Pareto principle are displayed in Table 5. With the lowest FPR and the maximum precision and accuracy of all these methods, the table demonstrates how well the A1DE model works. Due to its inability to effectively handle real-time datasets, the Hoeffding Tree model is less accurate.

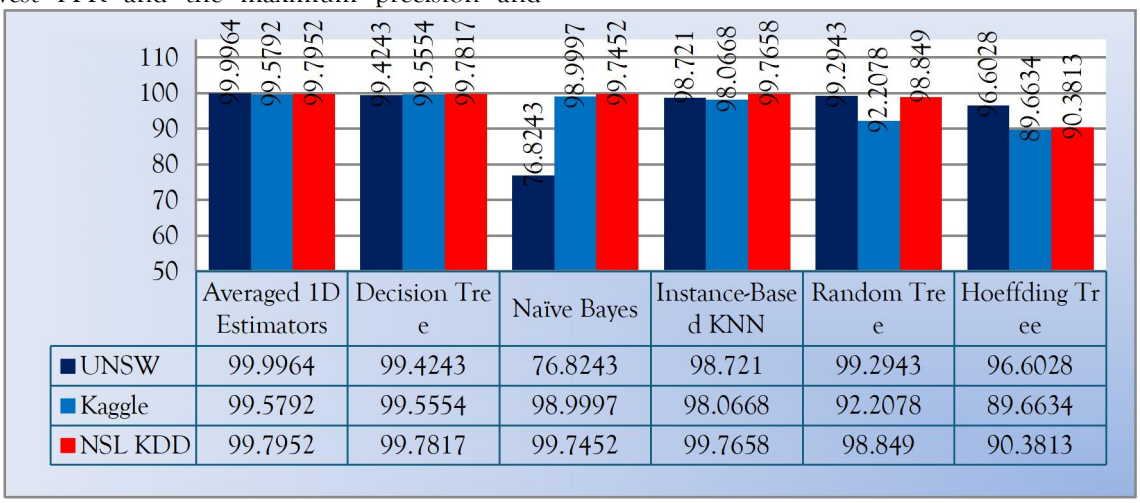
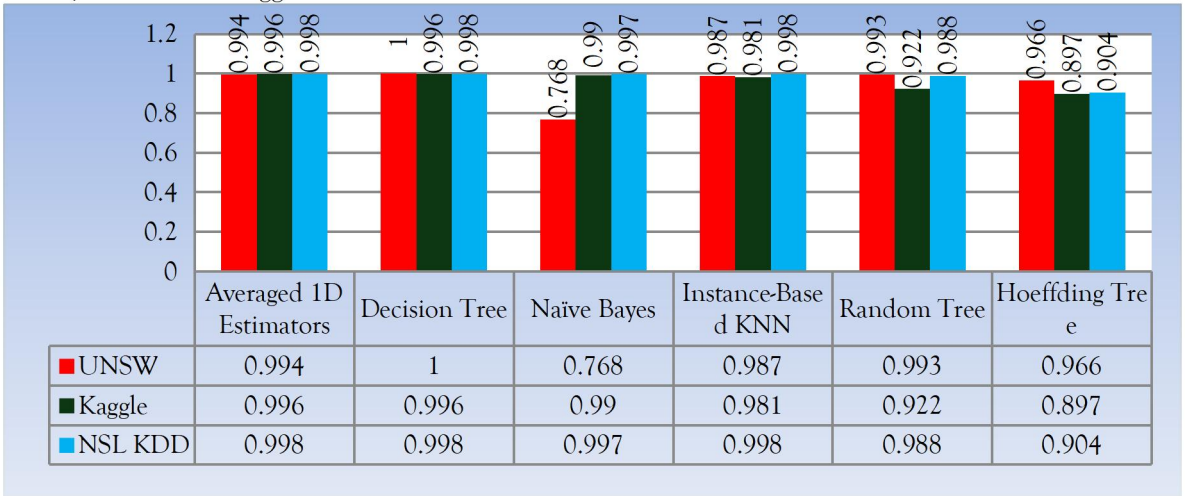


Figure 4. Accuracy Results of the Applied Techniques

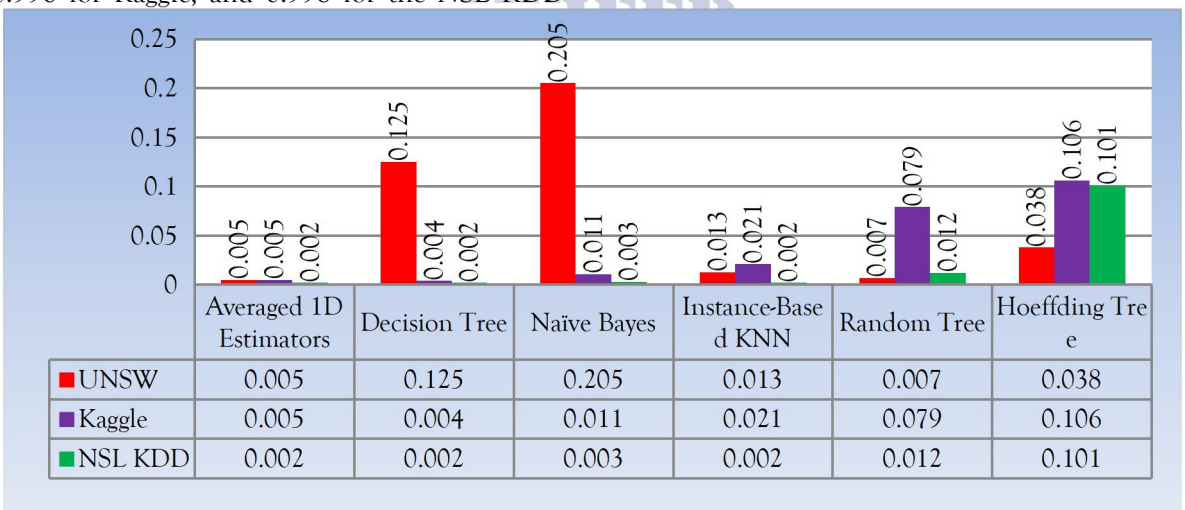
Fig. 4 shows the accuracy results of the applied techniques which are A1DE, DT, NB, IBK, RT, and HT model. From these results, we can conclude that the A1DE have highest accuracy rate, having 99.9964 accuracy rate for UNSW dataset, 99.5792 for kaggle and 99.7952 for NSL KDD dataset. Naives bayes model is less accurate for UNSW because it is real time and large sized dataset and the hoefdding tree also not performed well because the NSL KDD is multi classed dataset.



	Averaged 1D Estimators	Decision Tree	Naïve Bayes	Instance-Based KNN	Random Tree	Hoeffding Tree
UNSW	0.994	1	0.768	0.987	0.993	0.966
Kaggle	0.996	0.996	0.99	0.981	0.922	0.897
NSL KDD	0.998	0.998	0.997	0.998	0.988	0.904

Figure 5. True Positive Rate Metrics of the Applied Techniques

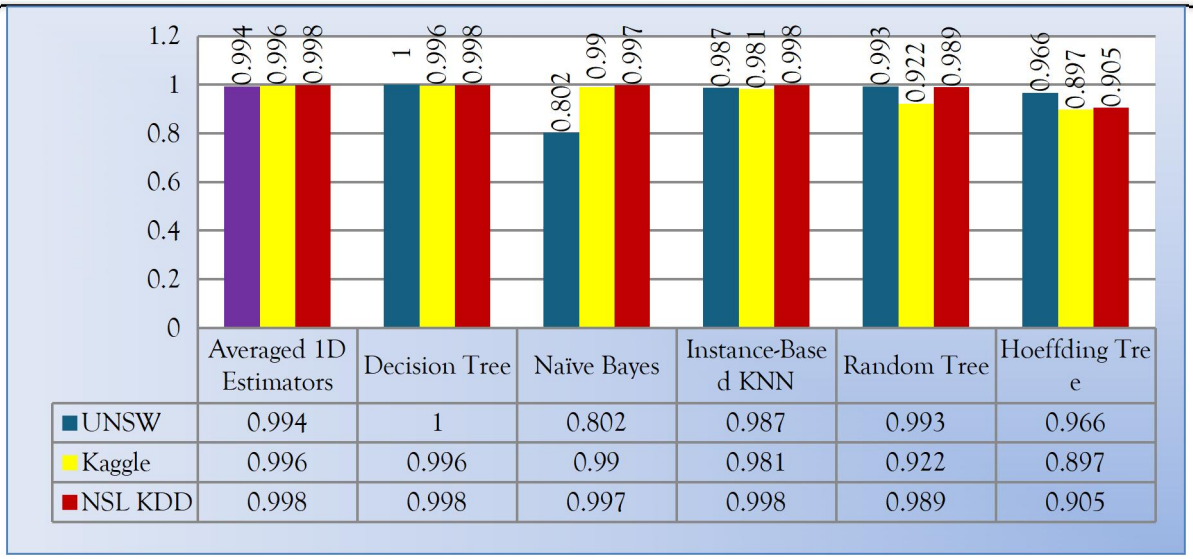
The true positive rate (TPR) outcomes of the implemented approaches—the A1DE, DT, NB, IBK, RT, and HT models—are displayed in Fig. 5. With a TPR rate of 1 for the UNSW dataset, 0.996 for Kaggle, and 0.998 for the NSL KDD dataset, we may conclude from these results that the Decision Tree has the greatest TPR rate. For two datasets, the A1DE and decision tree results are identical.



	Averaged 1D Estimators	Decision Tree	Naïve Bayes	Instance-Based KNN	Random Tree	Hoeffding Tree
UNSW	0.005	0.125	0.205	0.013	0.007	0.038
Kaggle	0.005	0.004	0.011	0.021	0.079	0.106
NSL KDD	0.002	0.002	0.003	0.002	0.012	0.101

Figure 6. False Positive Rate Metrics of the Applied Techniques

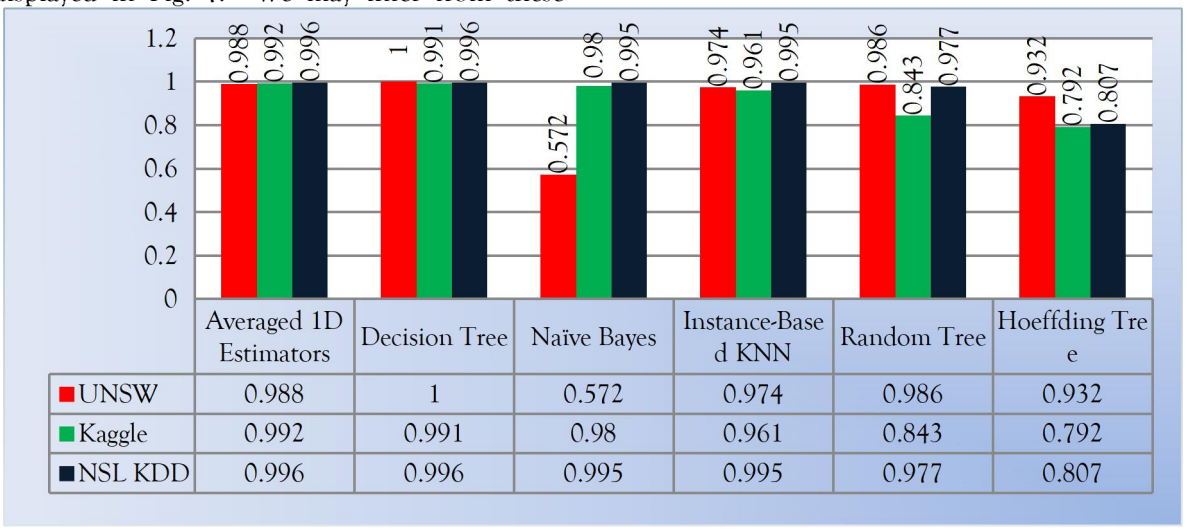
Fig. 6 shows the false positive rate (FPR) results of the applied strategies, which include the A1DE, DT, NB, IBK, RT, and HT models. We may conclude from these results that the A1DE has the lowest FPR rate, with an FPR rate of 0.005 for the UNSW dataset, 0.005 for Kaggle, and 0.002 for the NSL KDD dataset.



	Averaged 1D Estimators	Decision Tree	Naïve Bayes	Instance-Based KNN	Random Tree	Hoeffding Tree
UNSW	0.994	1	0.802	0.987	0.993	0.966
Kaggle	0.996	0.996	0.99	0.981	0.922	0.897
NSL KDD	0.998	0.998	0.997	0.998	0.989	0.905

Figure 7. Precision Results of the Applied Techniques

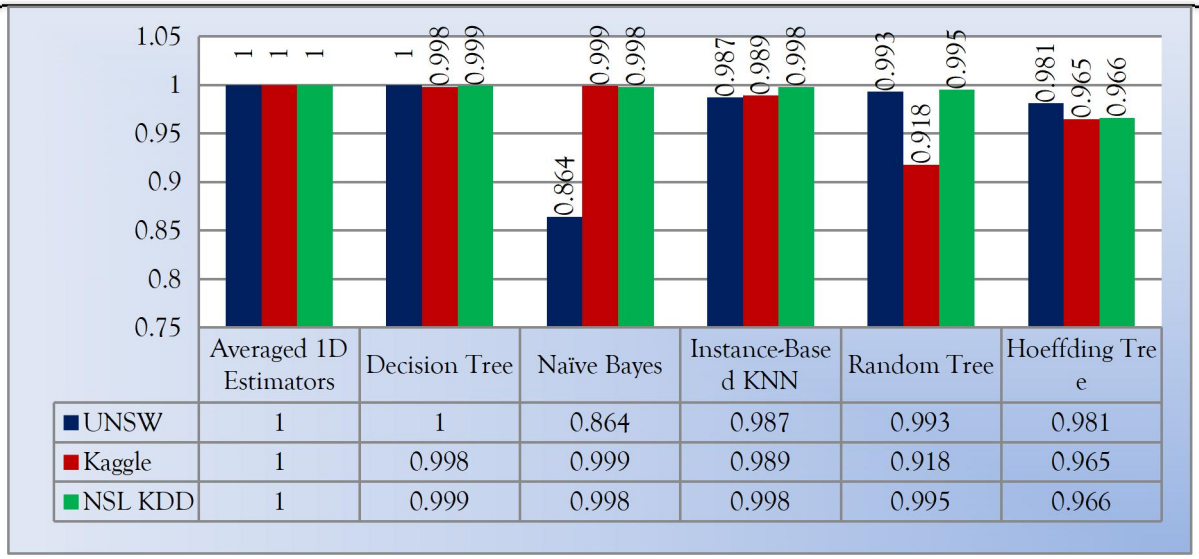
The precision outcomes of the used techniques—A1DE, DT, NB, IBK, RT, and HT model—are displayed in Fig. 7. We may infer from these statistics that the decision tree and A1DE have the highest.



	Averaged 1D Estimators	Decision Tree	Naïve Bayes	Instance-Based KNN	Random Tree	Hoeffding Tree
UNSW	0.988	1	0.572	0.974	0.986	0.932
Kaggle	0.992	0.991	0.98	0.961	0.843	0.792
NSL KDD	0.996	0.996	0.995	0.995	0.977	0.807

Figure 8. Mathews Correlation Coefficient Metrics of the Applied Techniques

The outcomes of Mathews Correlation Coefficient measures following the use of ML approaches such as A1DE, DT, NB, IBK, RT, and HT are displayed in Fig. 8. The A1DE performed well, according to the results, with an MCC rate of 0.998 for UNSW, 0.992 for Kaggle, and 0.996 for NSL KDD. When using decision trees, the MCC rate is likewise good for all datasets.



	Averaged 1D Estimators	Decision Tree	Naïve Bayes	Instance-Based KNN	Random Tree	Hoeffding Tree
UNSW	1	1	0.864	0.987	0.993	0.981
Kaggle	1	0.998	0.999	0.989	0.918	0.965
NSL KDD	1	0.999	0.998	0.998	0.995	0.966

Figure 9. ROC Area Results of the Applied Techniques

Fig. 9 shows the results of the ROC area after using ML techniques such A1DE, DT, NB, IBK, RT, and HT. The findings showed that the A1DE did well, with an MCC rate of 1 across all three datasets.

7. Discussion

Experimental investigation has demonstrated that, when applied to all three datasets—the NSLKDD, UNSW, and Kaggle—the A1DE-based intrusion detection model outperforms a number of popular machine-learning techniques. According to the findings, most algorithms performed better on distinct datasets; as a result, the patterns in the former data were more distinct. Nonetheless, A1DE showed excellent accuracy and low false-positive rates, demonstrating the validity and applicability of its findings in the context of all three datasets. The best method for detecting network intrusions is the A1DE model. It has a great track record of preventing false alarms, low false positive rates, and high detection rates, according to empirical investigations. The usefulness of ensemble approaches in network intrusion detection is highlighted by this type of superiority in ensemble learning scenarios. Unlike single base classifiers, A1DE combines many classifiers through aggregation, resulting in a stronger, less biased, and more accurate prediction. This minimizes excessive levels of bias and volatility. The model is a strong and effective tool for tackling contemporary cyber security concerns due to its high degree of performance across various data sets. It also serves as the foundation for the development of more sophisticated intrusion-detection systems.

8. Conclusion

In this paper, ML-based models which include A1DE, DT, NB, IBK, RT, and HT algorithms that are designed and tested for NIDS. Several performance metrics, such as accuracy, precision, TPR, and FPR, are taken into consideration for model efficiency, and three distinct datasets—UNSW, NSL-KDD, and Kaggle—are used for training and testing. The 80/20 concept is used to categorize the datasets, and performance measures are used to illustrate the outcomes of every approach. Nonetheless, when examining three datasets, A1DE consistently yields the best results for classifying normal and anomalous classes, which can help researchers, build a better NIDS. 99.9% accuracy is the maximum achieved by NIDS based on A1DE. The A1DE-based model outperformed the other applied approaches and the techniques listed in the literature review section, according to Tables 4, 5, and 6. The most cutting-edge methods, such as deep learning or graph neural networks, will need to be used on real-time datasets in the future. Additionally, IoT-based systems need to be used on multiclass datasets for real-time analysis.

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://cybersecurityventures.com/cybercrime-damages-6-trillion-by-2024/.
[11] 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.
[12] 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.
[13] 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.
[14] 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.
[15] 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.
[16] 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.
[17] 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.
[18] 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.
[19] 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.
[20] Jabbar, M. A., and Rajanikanth Aluvalu. "RFAODE: A novel ensemble intrusion detection system." Procedia computer science 115 (2017): 226-234.

[21] Baig, Zubair A., et al. "Averaged dependence estimators for DoS attack detection in IoT networks." Future Generation Computer Systems 102 (2020): 198-209.

[22] Saranya, T., et al. "Performance analysis of machine learning algorithms in intrusion detection system: A review." Procedia Computer Science 171 (2020): 1251-1260.

[23] Kasongo, Sydney M., and Yanxia Sun. "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset." Journal of Big Data 7.1 (2020): 105.

[24] Zakariah, Mohammed, et al. "Intrusion Detection System with Customized Machine Learning Techniques for NSL-KDD Dataset." Computers, Materials & Continua 77.3 (2023).

[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] A. Majid, J. H. Arman, F. Muhammad and B. Khan, "Enhancing Chronic Kidney Disease Diagnosis using Machine Learning Classifiers: A Comparative Analysis" 2nd International Conference on Contemporary and Academic Research (ICCAR), pp. 123-130, 2023.

[27] 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.

[28] Arman, J.H., , Fazal, M., Bilal K. and I. Khan, "Network Intrusion Detection System using Random Forest and Random Committee Models", International Conference on Internet of Things (ICIoT), pp. 18-25, 2022.

[29] 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.

[30] 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.

[31] 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.

[32] R. D. A. Khan, H. Ping, and M. Asif, "The impact of green human resource management on employee green performance through green commitment and transformational leadership," *Center for Management Science Research*, vol. 4, no. 5, pp. 635–677, May 2026, doi: 10.5281/zenodo.20510765.

[33] M. Asif, S. Karim, A. Latif, H. A. H. Asim, and A. Kareem, "Impact of behavioural biases on investment decisions: A study of individual investors in Pakistan," *Contemporary Journal of Social Science Review*, vol. 4, no. 1, pp. 1538–1550, 2026, doi: 10.63878/cjssr.v4i1.2578.

[34] M. Asif and M. Bashir, "Augmentation or Anxiety? The Mediating Role of Employee Trust in the Relationship Between Generative AI Implementation, Job Crafting, and Productivity," *The Critical Review of Social Sciences Studies*, vol. 4, no. 1, pp. 4550–4583, 2026, doi: 10.59075/mrqkn978.

[35] M. Rafiq-uz-Zaman and M. Asif, "Mechanisms of exclusion: Power, structure, and the persistence of gender inequality," *Qualitative Research Journal for Social Studies*, vol. 3, no. 1, pp. 690–703, 2026, doi: 10.63878/qrjs921.

[36] S. Ahmed and M. Asif, "Comparative analysis of attitudes toward climate change policies across urban and rural populations," *Pakistan Journal of Social Science Review*, vol. 5, no. 1, pp. 747–769, 2026, doi: 10.5281/zenodo.18457821.

[37] S. Ahmed and M. Asif, "Public opinion on the effectiveness of local government anti-corruption measures: A multi-city survey analysis," *International Journal of Social Sciences Bulletin*, vol. 4, no. 1, pp. 1189–1201, 2026, doi: 10.5281/zenodo.18412790.

[38] D. Mohiuddin, A. A. Zaveri, I. Ahmed, and M. Umar, "A systematic literature review of multi-channel analytics linked to POS and

connected to food businesses in the UK," in *2026 International Conference on AI Innovations and Industry (ICAIII)*, 2026, pp. 1–6. doi: 10.1109/ICAIII69475.2026.11521642.

[39] D. Mohiuddin, M. H. Tariq, and A. Tahir, "The Impact of Generative AI on Personalized Content Marketing in E-Commerce," *Inverge Journal of Social Sciences*, vol. 4, no. 1, pp. 162–188, 2025. doi: 10.63544/ijss.v4i1.288.