International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
Article Received: *25 April 2025* ***Revised:*** *12 May 2025****Accepted:*** *25 June 2025*

Machine Learning-based Intrusion Detection System for Social Network Infrastructure

Govind Kumar Jha

Department of Computer Science & Engineering, Government Engineering College Munger

Munger, India

gvnd.jha@gmail.com

Preetish Ranjan

Amity School of Engineering & Technology, Amity University Patna, Patna, India

pranjan@ptn.amity.edu

Ritesh Ravi

Amity Business School, Amity University Patna, Patna, India

rravi@ptn.amity.edu

Hardeo Kumar Thakur

Department of Computer Science & Engineering, Bennett University, Greater Noida, India

hardeo.thakur@bennett.edu.in

Abstract :

The growing number of cyber-attacks demands a critical measure to prevent unauthorized data access. Thus, intrusion detection has become critical to deal with such attacks. This work attempts to identify malicious connections using a few key parameters. The system has been trained using data relating to normal and abnormal events through machine learning and data mining techniques. To detect intrusions, this study assessed five distinct machine learning models: Random Forest, Bagging, Boosting, Support Vector Machine, and K-Nearest Neighbor (KNN). Based on the number of features, iterations, and hyperparameters, the models were evaluated using experimental data collected in real time. With a detection rate of up to 98.7%, the Random Forest approach surpassed existing machine learning models for intrusion detection. The paper proposes a novel intrusion detection system (IDS) based on these findings that successfully identifies possible threats before they seriously compromise network security and stop cyberattacks.

Index Terms: *Machine Learning, Intrusion Detection System, Random Forest, Support Vector Machine, K Nearest Neighbor (KNN), Bag and Boosting, Ensemble learning.*

1. Introduction

The term "intrusion" describes an unauthorized user's access to a system or network, frequently with malevolent intent. Even with sophisticated intrusion detection systems, firewalls frequently fail to identify the financial ramifications of assaults. Data breaches cost companies an average of $4.35 million in 2022, according to AAG IT Services (June 2023). Not to mention the harm to one's reputation and other losses, losing this much money in a cyberattack is a big worry. According to Petrosyan (2022), the global cost of cybercrime is increasing. Therefore, there is a great demand for cybersecurity products. Enterprises are swiftly creating Intrusion Detection Systems (IDS) in reaction to cyberattacks directed at both public and private organizations (Sultana, 2019). As a result of intrusions, there may be a rise in ransomware assaults, in

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
***Article Received:** 25 April 2025 **Revised:** 12 May 2025**Accepted:** 25 June 2025*

which a company's data is encrypted and rendered unreadable. These intrusions have major consequences if they are not discovered and examined promptly; for this reason, they must be handled carefully and strategically. Numerous methods, including network segmentation, firewalls, access control, behavioral analytics, data loss prevention, distributed denial of service (DDoS) prevention, antivirus, and anti-malware software, application security, and firewalls, are frequently used to prevent unauthorized access to the system. They have the ability to block data outflow, filter information, create alerts, and stop dangerous activity. In firewalls and spam filters, simple rule-based algorithms are frequently used to accept and reject protocols, ports, and IP addresses. However, firewalls and filters have limitations of distinguishing between 'good traffic' and 'bad traffic'.

Preparedness to deal with the consequences is of utmost importance. Therefore, this article aims to develop a model based on Machine Learning and Deep Learning techniques. The objective is to effectively distinguish between positive and negative connections, outperforming existing models while minimizing false positives. Figure 1 is the basic workflow of the model. The input traffic is fed to the trained model and this traffic will be either passed or blocked further based on characteristics of good or bad traffic.

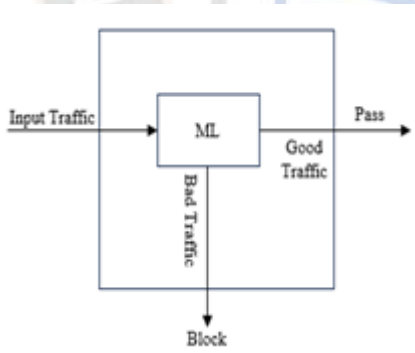


Figure 1. A basic model of the intrusion detection system

The application based on the proposed model will check the intrusion at every network level as shown in Figure 2. Point A and B is the place in any basic corporate network infrastructure where the proposed application can be deployed to filter the incoming traffic.

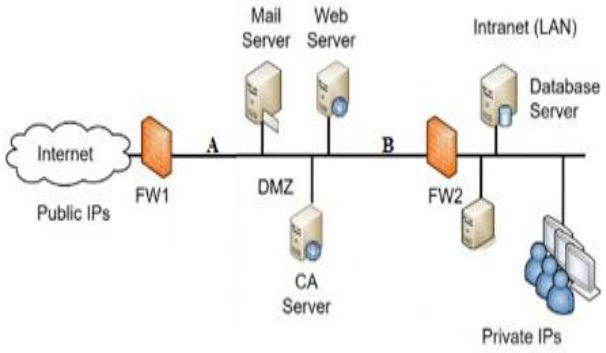


Figure 2. Corporate network infrastructure

The KDD cup 99 is the benchmark dataset that has been used to train and test the model (Bolon, 2011; Bhati, 2020). In their analysis, the authors (Aggarwal, 2015) examined the KDD dataset, specifically focusing on four categories: basic, content, traffic, and host. Similarly, (Norwahidayah, 2021) conducted research along similar lines, investigating the KDD Cup 99 dataset using Artificial Neural Networks (ANN) and Particle Swarm Optimization (PSO) techniques. However, the proposed model is more robust as it considers all the attributes of the dataset and normalizes the data with standard methodology.

The study assessed five distinct machine learning models—Random Forest, Bagging, Boosting, Support Vector Machine, and K-Nearest Neighbour (KNN)—in the context of intrusion detection. The models were tested on KDD dataset based on hyperparameters, the number of iterations, and the number of features. Random Forest technique outperformed other machine learning models for intrusion detection. The detection rate is more than 98.7 percent. Based on these results, the study proposes an intrusion detection system (IDS) that can effectively recognize potential threats before they cause severe damage to network security and prevent cyber-attacks.

The initial section of the paper presents an introduction, while the subsequent section provides an associated literature on IDS, the KDD dataset, and the methodology deployed in the proposed model. Then the following section presents the findings and discussion. The study concludes with the last section.

2. Literature Review

There is an extant of literature available on intrusion detection systems (IDS). Different machine learning-based IDS models have been proposed for various networks, such as computer networks, MANETS (Mobile Ad hoc Networks), WSN (Wireless Sensor Networks), Mobile Clouds, Internet of Things (IoT) Industrial Internet of Things (IIoT), Advanced Metering Infrastructure Networks, and SCADA (Supervisory control and data acquisition) Networks.

Sultana et al. (2018) conducted a review of the utilization of Software-Defined Networking (SDN) technology as a foundation for Machine Learning/ Deep Learning (ML/DL) based intrusion detection systems. The authors emphasized the significance of deep learning in assessing network security and highlighted the need to develop a feature selection method using classifiers to reduce dataset dimensions. The authors suggested that future research should focus on designing a centralized SDN controller for real-time intrusion detection in high-speed networks and applying SDN-based NIDS to critical infrastructure.

Amouri et al. (2020) suggested a multistage cross-layer intrusion detection system (IDS) based on machine learning for Mobile Ad hoc Networks (MANETS) and Wireless Sensor Networks (WSN). The model employed Iterative Linear Regression and Random Forest techniques and achieved a detection rate ranging from 90 to 98 percent. In addition to that Dey et al. (2019) proposed a machine learning-based IDS for mobile clouds, which can be customized according to the needs of heterogeneous client networks without requiring rule updates. The model involves two steps: decision-based Virtual Machine (VM) selection and multi-layer traffic screening, and it was highly effective in detecting intrusions.

Latif et al. (2020) developed a novel lightweight random neural network method for detecting cyber threats in the IIoT. Their proposed system exhibited excellent performance on the DS2OS dataset, consisting of seven distinct attack types. However, the authors noted that this dataset may not be comprehensive enough to fully evaluate the effectiveness of their approach for identifying threats in industrial IoT environments. The researchers discovered that their method outperformed conventional machine learning techniques like SVM, ANN, and decision tree on this dataset.

Haddad Pajouh et al. (2018) developed a technique that utilizes a recurrent neural network to detect malware in IoT devices. Their technique comprises data collection, feature extraction, and deep threat classifier. The study demonstrated that their approach outperformed other established machine learning classifiers such as Naive Bayes, K-Nearest Neighbor, Random Forest, and Decision Tree in terms of accuracy and efficiency.

Yihunie et al. (2019) conducted a study to determine the most efficient method for detecting anomaly traffic from the NSL-KDD dataset. The researchers experimented with diverse machine learning procedures, including Stochastic Gradient Decent, Random Forests, Logistic Regression, Support Vector Machine, and Sequential Model classifiers, and created a highly accurate classifier with a minimal error rate. The study revealed that the Random Forest Classifier was the most effective in detecting anomaly traffic, with or without normalization applied to the dataset.

Mrabet et al. (2019) proposed a deep learning-based intrusion detection system (IDS) for Advanced Metering Infrastructure (AMI) network using the NSL KDD dataset. Their method achieved an accuracy of 99.5 percent and outperformed other well-established machine learning classifiers like Random Forest, Naive Bayes, and Support Vector Machine. Ge et al. (2019) presented a feed-forward neural network-based IDS model for IoT networks, which achieved high accuracy in binary and multi-class classification, including denial of service, distributed denial of service, reconnaissance, and information theft attacks. Yang et al. (2019) proposed a convolutional neural network-based IDS for SCADA networks that demonstrated high detection accuracy and the ability to handle newly emerged threats.

Costa et al. (2019) conducted a literature survey on intrusion detection for IoT and found that the scientific community and industry are both focused on developing optimized security protocols to provide reasonable protection while maintaining low energy consumption. They reviewed various proposed IDS techniques to achieve better detection rates and noted that the false positive rate remains a problem. Abubakar and Pranggono (2017) developed a neural network-based IDS for SDN to detect anomaly-based attacks in the SDN environment. Their study aimed to improve IDS effectiveness for SDN by leveraging pattern recognition as the machine learning technique, which achieved high accuracy compared to other neural network models.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
***Article Received:** 25 April 2025 **Revised:** 12 May 2025**Accepted:** 25 June 2025*

Recently, Komal et al. (2024) explored some of IDS's most popular machine learning algorithms. They proposed a model namely REPOStack based on recursive feature elimination, self-adaptive equilibrium optimizer, Adaboost, support vector machine, deep neural network and XGBoost. They claimed promising results when applied over benchmark datasets such as NSLKDD, UNSW-NB15 and CICIDS [22]. In another work similar to Komal et al., Imran et al. proposed a hybrid feature selection technique composed of Pearson Correlation Coefficient and Random Forest model. They used TON_Iot dataset to train the machine through decision tree, AdaBoost, K-nearest neighbor and multilayer perceptron etc. Finally, they concluded that decision tree and multilayer perceptron provided optimal accuracy with few false positive and false negative results [23]. Intrusion is the main concern in the networked IoT devices, which causes limited use of IoT devices in various services. A report published by nature in 2024 outlined a two-stage procedure for determining and identifying intrusion in IoT network. It has been emphasized in the report that Extra Tree, Deep Neural Network and Random Forest techniques are producing improved accuracy and better stability [24]. It can be inferred from various studies that deep learning-based IDS models have better detection rates and can handle newly emerged threats. However, false positive rates remain a problem that needs to be addressed in future IDS. The use of software-defined networking (SDN) as a platform for intrusion detection using machine learning/deep learning approaches is also gaining attention, and there is a need for developing a centralized SDN controller for real-time intrusion detection in high-speed networks.

3. Dataset Specifications

The data set contains well-known parameters contributing to the classification of one connection/transaction as normal or anomaly. There are forty-two attributes referred to as features in the available data set. There are six nominal attributes including the class attribute which is to be predicted. The rest thirty-six attributes are numeric. Two datasets are available for training and testing, respectively. The training dataset, called train_set, is used in model building and contains 125974 instances. Another dataset, called test_set, is reserved for validation purposes and contains 22544 instances (around 20% of training instances). Both the datasets are perfectly balanced and with no missing, outliers are very suitable for binary classification.

4. Dataset Pre-Processing

Data pre-processing is essential to align and normalize the characteristics of the data for better performance of machine learning algorithms. The subsequent transformations have been used for the aimed work: one hot encoding for nominal features and standard normalization for numeric features.

Table 1. Dataset's characteristics

Dataset	# of total Instance	Class-wise output instances	
		Normal	Anomaly
train_set	125974	67344	58630
test_set	22544	9711	12833

The concerned dataset does not have any missing values, data imputation is not needed. For data standardization, Z-score normalization is used which manipulates mean and standard deviation of the attribute.

Considering, M_i and S_i as the mean and standard deviation of the i^{th} attribute F_i of the concerned dataset, D, then the z-score N_{ij} value of element x_{ij} for j_{th} instance I_j is determined as shown by Equation 1.

$$Nij = \frac{Xij - Mi}{Si} \qquad (1)$$

where M_i is calculated for attribute S_i as given in Equation 2.

$$Mi = \frac{1}{|I|}\sum_{k=1}^{|I|} Xk \qquad (2)$$

where |I| represents the total instance count for attribute i.

The data set division into training and validation sets is done in a stratified manner. This helps in maintaining the proportion of instances per class in training and

validation sets. The train and validation ratio are maintained at around 80:20.

5. Proposed methodology

The training and testing datasets are preprocessed in this proposed work as presented in IV Section. The datasets are partitioned in a ratio of 80:20 along with stratification.

As part of our procedure (Figure 3), concerning the training set, in the first epoch for 2 random features, we iterate 1 to 50 iterations and keep the results. After 50th iteration, iterate at the interval of 10 and after 100th iteration take iteration interval of 100 and keep these results also. This 2nd step is just to identify uniqueness of pattern of accuracy. Afterwards, we select one best result (more promising values of performance metrics described in previous subsection) and the very first iteration at which that result was obtained. Similar procedure is repeated for all other epochs (comprising of 3,4,5, and continued till 41 numbers of random features) to obtain one best result (as discussed above). Table 2 clearly shows these results obtained with our experimentation.

This is a binary classification problem. We consider ML algorithms <*Random Forest, Bagging*> with default set of hyper-parameters in the python environment. The performance metrics is <*Accuracy, FPR, Precision, TPR/Recall/Sensitivity, F1-Score, MCC, ROC-AUC, Kappa*> and the most concerned element of metrics is *Accuracy*, along with *Sensitivity*, *Specificity*, and *Kappa*. With this algorithm, we were able to find the best combination of model, performance metrics, and the number of features.

Table 2. Performance evaluation metrics for RF, Bagging, Boosting, SVM, and KNN classifiers

Classifier	Hyper-parameter	TP	TN	FP	FN	Specificity	Sensitivity	Precision	F1-Score	MCC	AUC	Kappa	Accuracy
RF	criterion : entropy; weight:Defualt estimators:100	13457	11705	21	12	0.9982	0.9991	0.9990	0.9988	0.9974	1.0000	0.9982	99.87
Bagging	estimators : 100	13444	11704	22	25	0.9981	0.9981	0.9980	0.9983	0.9963	0.9999	0.9970	99.81
Boosting	estimators : 100	13348	11565	161	121	0.9863	0.9910	0.9990	0.9895	0.9775	0.9992	0.9981	98.88
SVM	kernel : poly	13351	11527	199	118	0.9830	0.9912	0.9811	0.9883	0.9747	0.9992	0.9877	98.74
KNN	neighbors : 5	13423	11673	53	46	0.9955	0.9966	0.9889	0.9863	0.9921	0.9992	0.8921	99.61

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
Article Received: *25 April 2025* ***Revised:*** *12 May 2025****Accepted:*** *25 June 2025*

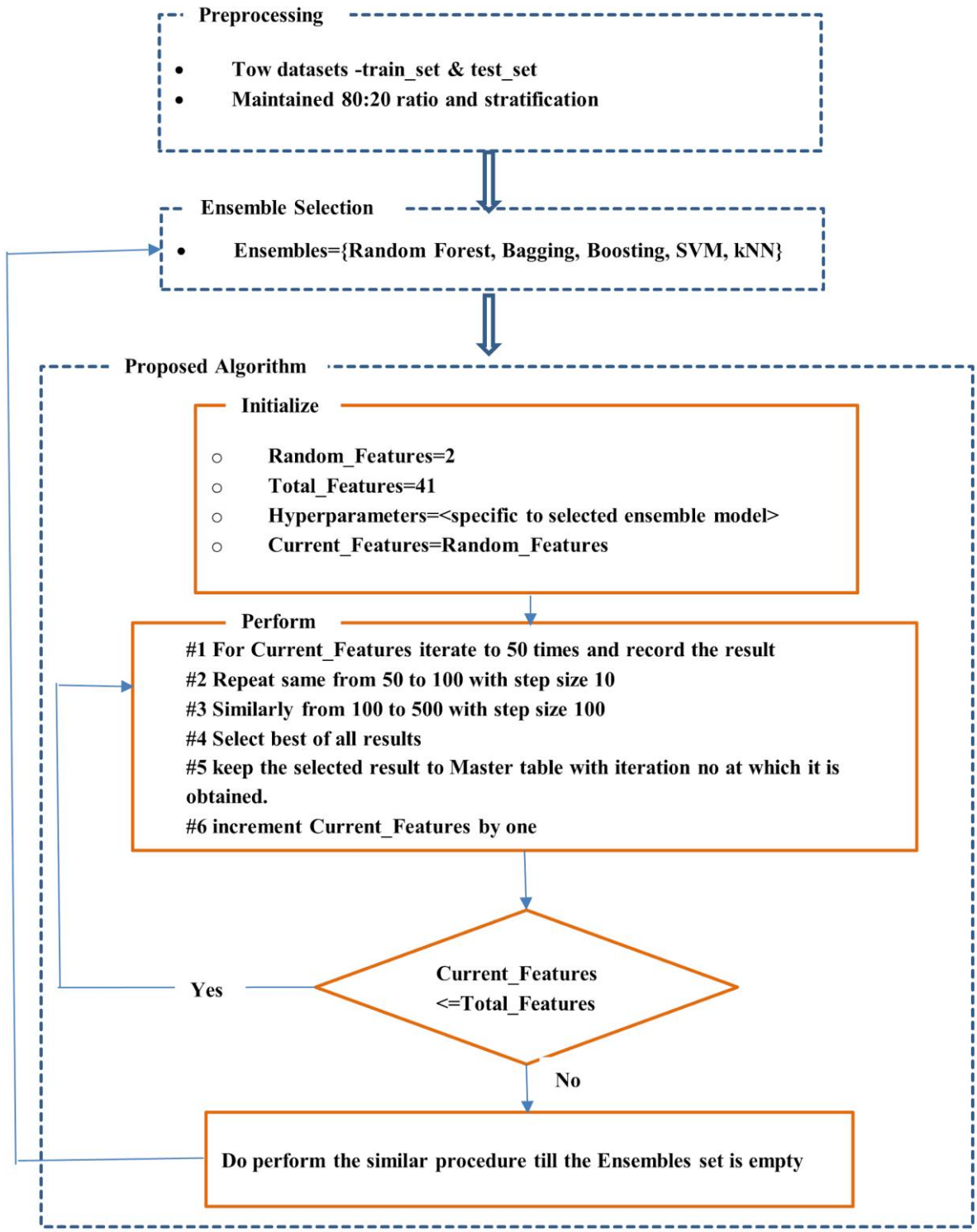


Figure 3. Flow Chart of the proposed methodology and algorithm

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
Article Received: *25 April 2025* ***Revised:*** *12 May 2025****Accepted:*** *25 June 2025*

6. Results Analysis and Discussion

The experimentation detail and performance evaluation are concluded in this section. Section A lists the tools and equipment used for experimentation. Section B presents the detailed evaluation criteria and performance metrics used. Finally, section C summarizes the analysis of the results obtained.

A. Experimental Setup

We have used Python environment 3.7.1 in Windows 10 OS with hardware configuration – Intel Core i5 CPU @ 1.60 GHz, 8 GB RAM.

B. Evaluation Criteria

A confusion matrix is obtained, and various performance metrics are calculated using the components of the confusion matrix- TP, TN, FP, FN. TP refers to the number of correct/true instances which are predicted correct/true. TN refers to number of incorrect/false instances which are predicted incorrect/false. The cross reference between true and false instances gives notion to FP and FN.

In this work we do binary classification. The confusion matrix is used to evaluate all metrics including- *Accuracy, Precision, TPR/Recall/Sensitivity, F1-Score, MCC, ROC-AUC, Kappa*.

The accuracy (*Accuracy*) represents the accurate classification done and is calculated mathematically through confusion matrix components as given in Equation 4.

Precision (Equation 5) is the accuracy of the positive prediction (Geron,2019). Recall represents ratio of positive prediction to actual positives (Geron, 2019). The F-1 score is a harmonic mean of precision and recall. A high F-1 score will result if both precision and recall are high (Geron, 2019). The ROC curve is a plot between TPR and FPR. The FPR is the ratio of negative instances that are incorrectly classified as positive. It is equal to 1-TNR/Specificity (Geron, 2019). ROC-AUC curve accurately signifies the amount of separation between the classes. High AUC means the high capability of the model to distinguish true class as true class and false class as false class. TNR/specificity is the ratio of negative instances that are being correctly classified as negative.

The metrics Sensitivity, Specificity, and F1 Score are described mathematically in terms of confusion matrix components as given in Equation 5-7, respectively.

$$Precision = \frac{TP}{TP+FP} X\ 100\% \quad (3)$$

$$Accuracy = \frac{TN+TP}{TP+FP+FN+TN}\ X\ 100\% \quad (4)$$

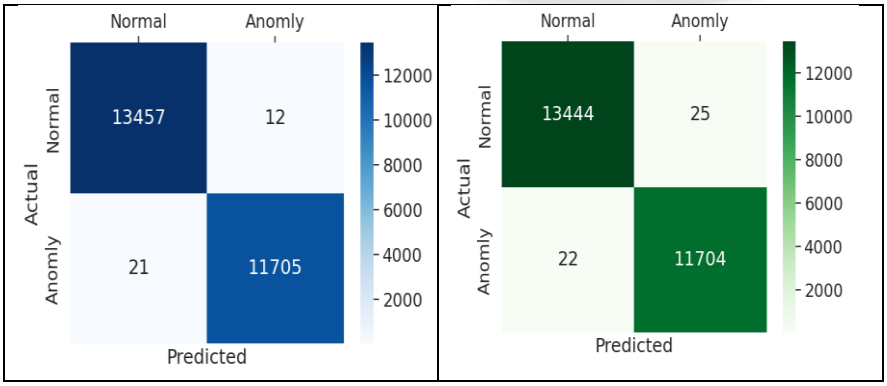
$$Sensitivity = \frac{TP}{TP+FN} X\ 100\% \quad (5)$$

$$Specificity = \frac{TN}{FP+TN}\ X\ 100\% \quad (6)$$

$$F1\ Score = \frac{2\ XTP}{2\ X\ TP\ X\ FP\ X\ FN}\ X\ 100\% \quad (7)$$

Mathew's correlation coefficient (MCC) metric is used to predict the classification score ranging between [-1, +1]. The values +1, -1 and near to zero indicate the ideal, completely wrong and random predictions, respectively (Equation 8).

$$MCC = \frac{TP\ X\ TN - FP\ X\ FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}\ X\ 100\%$$



International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
Article Received: *25 April 2025* ***Revised:*** *12 May 2025****Accepted:*** *25 June 2025*

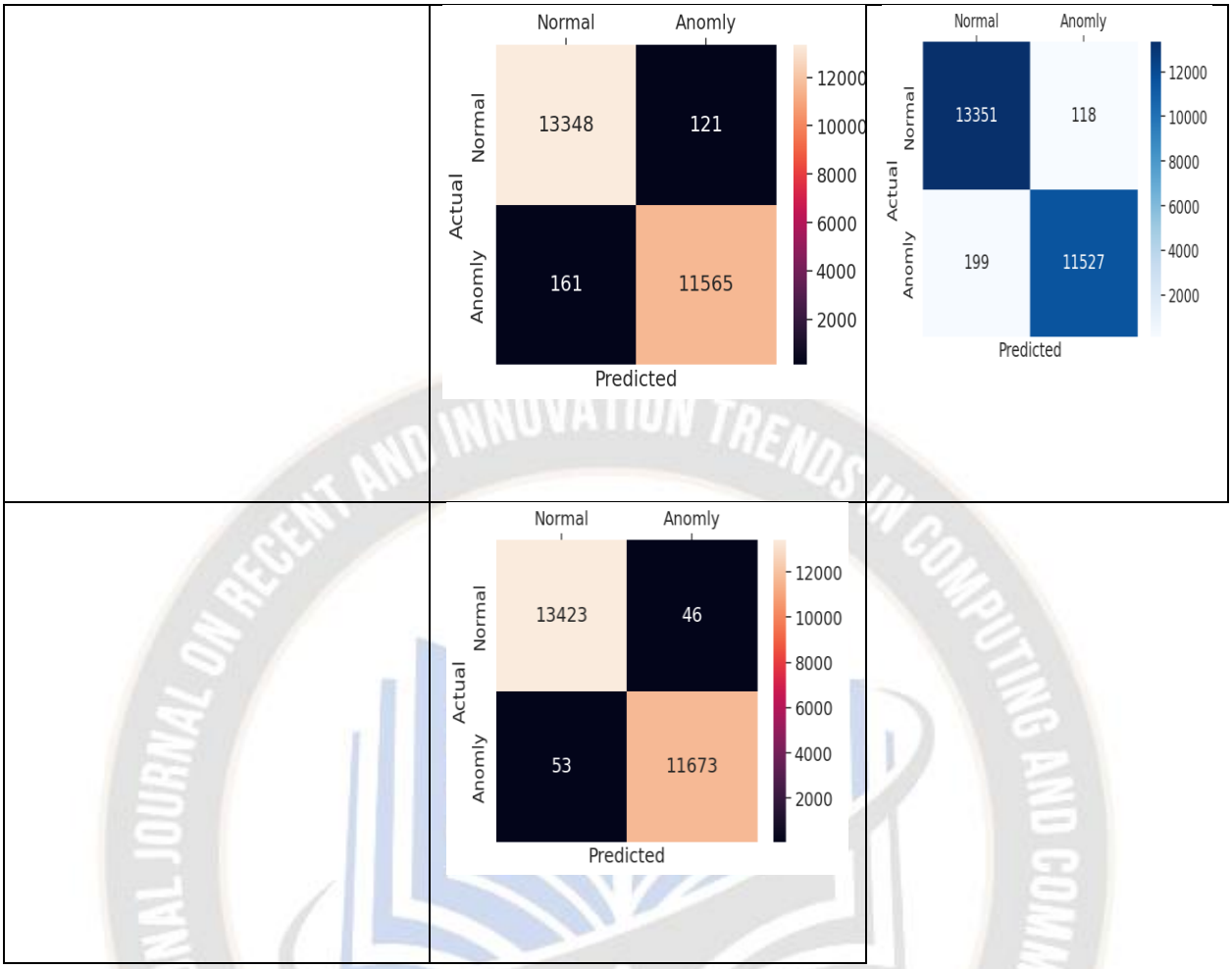


Figure 4. Confusion matrices (a) RF (b) Bagging (c) Boosting (d) SVM (e) kNN

Kappa measure tells how much better the classifier is performing and this can be defined as, $Kappa = \frac{Po-Pe}{1-Pe}$, where *Po* is observed agreement and *Pe* is expected agreement. Landis and Koch (1977) described, the range of kappa values and their interpretation is – <0 [no agreement], 0 - 0.20 [slight], 0.21 - 0.40 [fair], 041 - 0.60 [moderate] , 0.61 - 0.80 [substantial] and 0.81 - 1 [almost perfect].

C. Performance Evaluation of Proposed Work

Best Performing Model

It can be analyzed that Random Forest is optimal binary classifier in terms of *accuracy* and *kappa* while exploiting minimum eight & maximum 32 features. *Accuracy* achieved 99.87% and *kappa* is 0.9982 [almost perfect]. The minimum number of iterations to achieve this accuracy is eight and to achieve the same without fail is 32 and beyond. Other metrics measurement for *Random Forest are Precision = 0.999, Recall = 0.9991, F1 Score = 0.9988, MCC = 0.9974.*

Slightly less performance is achieved by the Bagging approach where *Accuracy = 99.81%* and *Kappa= 0.9970* but the best part is that this performance is achieved by using just a minimum five number of features. Consistently, this performance achieved with number of iterations 36 and beyond. The other metric scores for bagging are *Precision = 0.998, Recall = 0.9981, F1Score = 0.9983, MCC=0.9963.*

Best kappa

Best Kappa As this is a relatively less imbalanced classification, hence kappa statistics can be a good measure to determine the performance of the random guessing model. The kappa statistics for the *<RF, Bagging>* are *<0.9982,0.9970>* showing *RF Model* superior over *Bagging* although both measurements fall in the [almost perfect] category.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
Article Received: *25 April 2025* ***Revised:*** *12 May 2025****Accepted:*** *25 June 2025*

Specificity & Sensitivity

Sensitivity is the ratio of actual positives out of total actual positives and a value close to 1 is desirable. for the *<RF, Bagging>* the observed and stabilized measurement for this metric is *<0.9982,0.9981>*. Similarly, specificity is a measurement as the ratio of actual negatives out of total actual negatives. Consequently, this should also be very near to 1. Our result shows the more promising value for RF compared to that of Bagging. This measurement is *<0.9991,0.9981>* for RF and Bagging respectively. Coincidentally, these results for *Sensitivity and Specificity* are obtained with eight number of features and five number of features of *RF and Bagging* respectively.

Best MCC-AUC Score

This measurement represents a strong correlation between actual and predicted values and ranges between -1 to +1. RF shows considerably good score i.e. 0.1 compared to *Bagging* which scores *0.999*.

Precision & F1-Score

A classification measure, Precision is sometime very useful. High precision classifiers are preferable over low precision classifier hence our results also favor RF classifier against Bagging with the scores *0.999* and *0.998* respectively. But Precision alone is not appropriate and must be accompanied with Recall to give a single and more powerful measurement *F1-Score*. In our result RF has a high F1-Score equal to *0.9988* while *Bagging* has low equal to *0.9983*.

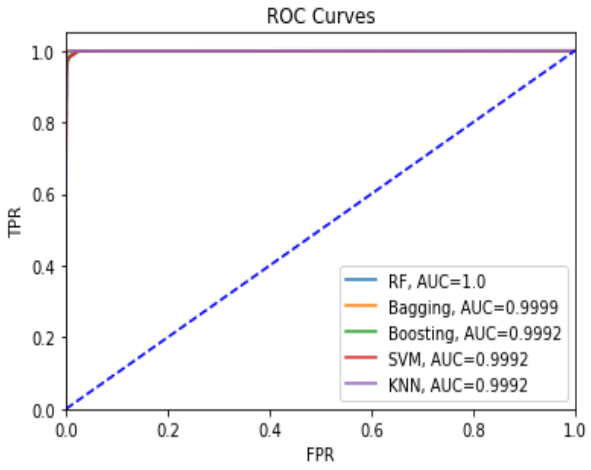


Figure 5. ROC_AUC Plots

Performance Representation through ROC_AUC Curve

ROC_AUC scores are used to denote the capability of model to distinguish among classes. For Random Forest and Bagging, the ROC_AUC score are *1.000 and 0.999* respectively. Figure 5 depicts all classifier's ROCs.

Performance Representation through Confusion Matrix

A close observation of confusion matrices (Figure 4) of all the classifiers states that Random Forest and Bagging approaches nearly match the performance in this binary classification. For *"Normal*" and "*Anomaly*" classes, the number of correctly classified instances is similar in both the classifiers whereas the bagging does more wrong classification than the Random Forest for the "Normal" class.

A comparison with other's work

Rahila et.al has also analysed the same dataset with different sets of algorithms such as SVM, Decision Tree, Random Forest, Ada Boost and Logistic Regression. They also reduced the dimension of dataset though feature selection algorithm. They concluded the highest accuracy of 99.5% with Random Forest (Rahim et al, 2022). The proposed model in this paper signifies that when bagging (ensemble) model is used over dataset then the accuracy level climbs up to 99%. Sherin et. al (2022) compared their model with existing Hybrid ML model having binary classification produced the accuracy of 90.4%. The proposed model in this paper is also having binary classification without selective attributes which makes the system more robust.

7. Conclusion

This study explores the utility of machine learning algorithms for IDS in today's networked environments. The well-known KDD dataset is analyzed, preprocessed, and experimented with a novel methodology (algorithm) to find the best-performing ML model among a variety of popular models available these days. A set of ML algorithms including instance-based (kNN, SVM), structure-based (Random Forest), and Ensemble methods (Bagging, Boosting) is used. This study reveals that the ensemble methods outperform the other models while exploiting the most features from the dataset. With this study, we concluded that the RF has the highest accuracy over the others which is quantified as 99.87%.

The superiority of this algorithm is also measured in different performance parameters including kappa, precision, recall, and MCC. The study serves as a good framework for new researchers in KDD analysis for IDS using traditional ML. In the future, Deep Learning methods can also be utilized to have better performance and in turn more robust IDS.

Compliance with Ethical Standards

All authors declare that they have no conflict of interest, financial or otherwise.

Funding Information

There is no funding provided in the course of this study.

References

[1] A. Petrosyan, "Estimated cost of cybercrime worldwide from 2016 to 2027," 2022.

[2] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on sdn based network intrusion detection system using machine learning approaches," Peer-to-Peer Networking and Applications, vol. 12, pp. 493–501, 2019.

[3] Martino, L; Paya Santos, C. A. & Delgado Morán, J, J. (2024). Thus,do they all: APTs as instruments of State-Sponsored cyber operations. Eksploriump. V. 45 No. 1s, 27-50. https://doi.org/10.52783/eksplorium.145

[4] V. Bolon-Canedo, N. Sanchez-Marono, and A. Alonso- Betanzos, "Feature selection and classification in multiple class datasets: An application to kdd cup 99 dataset," Expert Systems with Applications, vol. 38, no. 5, pp. 5947–5957, 2011.

[5] B. S. Bhati and C. Rai, "Ensemble based approach for intrusion detection using extra tree classifier," in Intelligent Computing in Engineering: Select Proceedings of RICE 2019. Springer, 2020, pp. 213–220.

[6] P. Aggarwal and S. K. Sharma, "Analysis of kdd dataset attributes-class wise for intrusion detection," Procedia Computer Science, vol. 57, pp. 842–851, 2015.

[7] S. Norwahidayah, N. Farahah, A. Amirah, N. Liyana, N. Suhana et al., "Performances of artificial neural network (ann) and particle swarm optimization (pso) using kdd cup '99 dataset in intrusion detection system (ids)," in Journal of Physics: Conference Series, vol. 1874, no. 1. IOP Publishing, 2021, p. 012061.

[8] A. Amouri, V. T. Alaparthy, and S. D. Morgera, "A machine learning based intrusion detection system for mobile internet of things," Sensors, vol. 20, no. 2, p. 461, 2020.

[9] S. Latif, Z. Zou, Z. Idrees, and J. Ahmad, "A novel attack detection scheme for the industrial internet of things using a lightweight random neural network," IEEE Access, vol. 8, pp. 89 337–89 350, 2020.

[10] H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K.-K. R. Choo, "A deep recurrent neural network based approach for internet of things malware threat hunting," Future Generation Computer Systems, vol. 85, pp. 88–96, 2018.

[11] F. Yihunie, E. Abdelfattah, and A. Regmi, "Applying machine learning to anomaly-based intrusion detection systems," in 2019 IEEE Long Island Systems, Appli- 6 cations and Technology Conference (LISAT). IEEE, 2019, pp. 1–5.

[12] Z. El Mrabet, M. Ezzari, H. Elghazi, and B. A. El Majd, "Deep learning-based intrusion detection system for advanced metering infrastructure," in Proceedings of the 2nd international conference on networking, information systems & security, 2019, pp. 1–7.

[13] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, and A. Robles- Kelly, "Deep learning-based intrusion detection for iot networks," in 2019 IEEE 24th pacific rim international symposium on dependable computing (PRDC). IEEE, 2019, pp. 256–25 609.

[14] H. Yang, L. Cheng, and M. C. Chuah, "Deep-learning based network intrusion detection for scada systems," in 2019 IEEE Conference on Communications and Network Security (CNS). IEEE, 2019, pp. 1–7.

[15] K. A. Da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of things: A survey on machine learning-based intrusion detection approaches," Computer Networks, vol. 151, pp. 147– 157, 2019.

[16] A. Abubakar and B. Pranggono, "Machine learning based intrusion detection system for software defined networks," in 2017 seventh international conference on emerging security technologies (EST). IEEE, 2017, pp. 138–143.

[17] A. Géron, Hands-on machine learning with Scikit- Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, 2019.
[18] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," biometrics, pp. 159–174, 1977.
[19] R. Rahim, A. S. Ahanger, S. M. Khan, and F. Ma, "Analysis of ids using feature selection approach on nslkdd dataset," 2022.
[20] V. I. J. Sherin and N. Radhika, "Stacked ensemble ids using nsl-kdd dataset," Journal of Pharmaceutical Negative Results, pp. 351–356, 2022.
[21] Preetish Ranjan, Vrijendra Singh, Prabhat Kumar, Satya Prakash, "Models for the detection of malicious intent people in society" International Journal of Digital Crime and Forensics (IJDCF), pp. 15-26, 2018.
[22] Preetish Ranjan, Abhishek Vaish, "Apriori Viterbi model for prior detection of socio-technical attacks in a social network", International Conference on Engineering and Telecommunication, pp. 97-101, 2014.
[23] Gill, K.S., Dhillon, A, "A hybrid machine learning framework for intrusion detection system in smart cities", Evolving Systems, 2024.
[24] Imran Hidayat, Muhammad Zulfiqar Ali, Arshad, "Machine Learning-Based Intrusion Detection System: An Experimental Comparison", Journal of Computational and Cognitive Engineering, Vol 2(2), 2023.
[25] Abdulaziz Aldaej, Imdah Ullah, Tariq Ahamed Ahanger, Mohammed Aliquzzaman, "Ensemble Technique of Intrusion Detection for IoT-edge platform", A scientific report published in nature, 2024.