Intrusion Detection using Ensemble Machine Learning

Ms. Nikita Kotangale[1], Dr. Shrikant Sonekar[2], Dr. Supriya S. Sawwashere[3], Prof. Mirza Moiz Baig[4]

[1]Ms. Nikita Kotangale, Computer Science & Engineering, JDCOEM
[2]Dr. Shrikant Sonekar, Computer Science & Engineering, JDCOEM
[3]Dr. Supriya S. Sawwashere, Computer Science & Engineering, JDCOEM
[4]Prof. Mirza Moiz Baig, Computer Science & Engineering, JDCOEM

Abstract- Now a days intrusion detection systems are essential for defending computer networking toward hostile activity. With the increasing complexity and diversity of modern cyber threats, traditional single-classifier-based IDS approaches often struggle to achieve optimal detection performance. To address this challenge, this study proposes an Intrusion Detection System using Ensemble Machine Learning. The methodology combines the strengths of multiple machine learning algorithms in an ensemble framework to enhance the accuracy, robustness, and efficiency of intrusion detection. The system incorporates steps such as data preprocessing, feature selection, ensemble model construction, and model performance. Techniques like data balancing, attribute encoding, and feature selection based on correlation are applied to optimize the IDS performance. The ensemble model benefits from the collective intelligence and diverse decision-making of multiple classifiers, improving the system's ability to accurately identify and respond to network intrusions. Through comprehensive result analysis, the study validates the effectiveness of the proposed IDS in terms of evaluation metrics, feature importance, robustness, and real- world impact. The proposed IDS using Ensemble Machine Learning offers a promising approach to tackle the dynamic and evolving nature of cyber threats, enhancing the security and resilience of computer networks.

Keywords *- Intrusion Detection System, Ensemble Machine Learning, Data Balancing, Feature Selection,* Cyber Security.

INTRODUCTION

An intrusion detection system (IDS) is a crucial component of network security that helps protect computer systems and networks from unauthorized access or malicious activities. Traditional IDS solutions typically rely on signature-based detection methods, where known patterns or signatures of attacks are compared against network traffic to identify potential threats. However, with the evolving nature of cyber threats, signature-based detection alone may not be sufficient [1].

Because allowing intrusion detection platforms to recognize and response to attacks that had not yet been seen or understood, machine learning (ML) has become a potent tool for improving intrusion detection platforms. ML-based IDS systems leverage algorithms that can learn from data and automatically detect patterns or anomalies indicative of malicious activities [2].

The use of machine learning in intrusion detection offers several advantages. Firstly, ML algorithms can analyze large volumes of network data in real time, enabling quick detection of anomalies or attacks that might go unnoticed by manual inspection. ML models can also adapt to evolving threats by continuously learning from new data, making them more robust in detecting novel attack patterns [3]-[4].

The detection of anomalies and abuse recognition are the two primary categories for machine learning methods utilized in IDS. Anomaly detection focuses on identifying deviations from normal behavior in network traffic, whereas misuse detection involves comparing network activity against pre-defined attack patterns or rules [5]-[6]. Some ML techniques commonly used in IDS include logistic regression, gradient boost, random forest, clustering algorithms, and ensemble approach.

To develop an ML-based IDS, a training phase is typically required, where the algorithm is exposed to a labeled dataset

DOI: 10.55041/IJSREM36806

consisting of both normal and malicious network traffic. This dataset helps the algorithm learn the characteristics of different types of attacks [7]. Once the model is trained, it can be deployed in a production environment to monitor and analyze incoming network traffic in real time, flagging any suspicious or potentially malicious activity.

However, it's important to note that ML-based IDS systems are not foolproof and can encounter challenges. They may produce false positives or false negatives, where normal traffic is misclassified as malicious or vice versa. Adversarial attacks that aim to deceive ML models can also pose a significant challenge.

RELATED WORK

A number of intrusion detection methods were used in the development of Intrusion IDS. Mishra et al. [8] suggested the techniques the mathematical, machine learning-based, as well as information mining techniques. Many research has used a single technique in these techniques, while some have selected to combine techniques to improve the efficacy of intrusion identification [9]. Abnormal recognition based on profiling signature was created by Atefi et al. [10] utilizing genetic algorithms and support vector machine methods. Regarding precision score, SVM worked better than GA. The two algorithms were integrated by the investigators to create a hybrid IDS. The effectiveness of the hybridized IDS was superior to the individual techniques in the assessment. The use of three different ensemble learning algorithms for abnormalities Detection was examined by Khoei et al. [10]. Bagging, boosting, and stacking were the three approaches used. The three approaches' effectiveness was evaluated to those of DT, NB, and KNN. The findings demonstrated that, in regard to detection rates, false alarm rates, missed detection rates, and accuracy scores.

An intrusion detection framework based on SVM and RF techniques was created by Rakshe et. al. [11]. The two techniques were employed to categorize data. The standard NSL-KDD was used to examine the performance of models. The models achieved greater than 96% accuracy in attack detection. When the two classifiers' performances were compared, the RF approach outperformed SVM in classifying traffic. An intrusion detection method was created by Kumar et al. [12] using four strategies such as NB, ID3, MLP, and ensemble learning. The standard benchmark CICIDS2017 dataset was used to assess the classifiers. The NB, ID3, and MLP were combined to create the ensemble model. Precision, recall, accuracy, and F1 score were the measures employed in the assessment of the classifiers. In comparison to the remaining models, ID3 (DT) worked well. In [13] describes how feature extracting and classifying approaches were integrated to improve detection rates whereas lowering false alarm rates. Chi-square was employed in the first step of feature identification. The aim of this phase was to keep the important characteristics that identify attacks while reducing the overall numbers of attributes in the dataset. A multiclass SVM technique was applied for categorization in the subsequent phase. This approach employed a multiclass SVM to increase categorization accuracy. The NSL-KDD dataset was used in assessing the framework, and the findings showed that the framework had a significant detection score and a low false alarm score.

A hybrid intrusion detection framework composed of convolutional and RNN has been suggested by Khan. The research's objective was to improve feature processing, that is crucial for intrusion identification technologies to function well. In the first phase, the localized features of the datasets were retrieved using CNN, and in the second phase, the geographic features were obtained by employing RNN. This approach addressed the issue of data imbalance in the dataset that was presented. The CICIDS 2018 dataset, regarded as the most recent dataset, was utilized to evaluate the strategy's effectiveness. With an effectiveness rate of 97%, the framework exceeded existing intrusion detection algorithms [14].

PROPOSED METHODOLOGY

identification, the suggested approach for an IDS employing ensemble classifiers integrates the advantages of various machine learning techniques. The system utilizes an ensemble approach, where multiple classifiers are trained on different subsets of the available data. These classifiers can include techniques such as logistic regression, decision trees, random forests, KNN, and Naïve Bayes. The ensemble model increases the identification probability and decreases the false positive percentage by pooling the forecasts obtained from these separate classifiers. In order to reduce the complexity of the information and increase the effectiveness of the entire system, the approach also uses feature selecting strategies to choose the most pertinent attributes for identifying intrusions. The proposed IDS methodology offers a robust and comprehensive approach to detecting and mitigating network intrusions, ensuring the security and integrity of computer systems and networks.

Fig. 1. Workflow of Proposed Network Intrusion Detection System

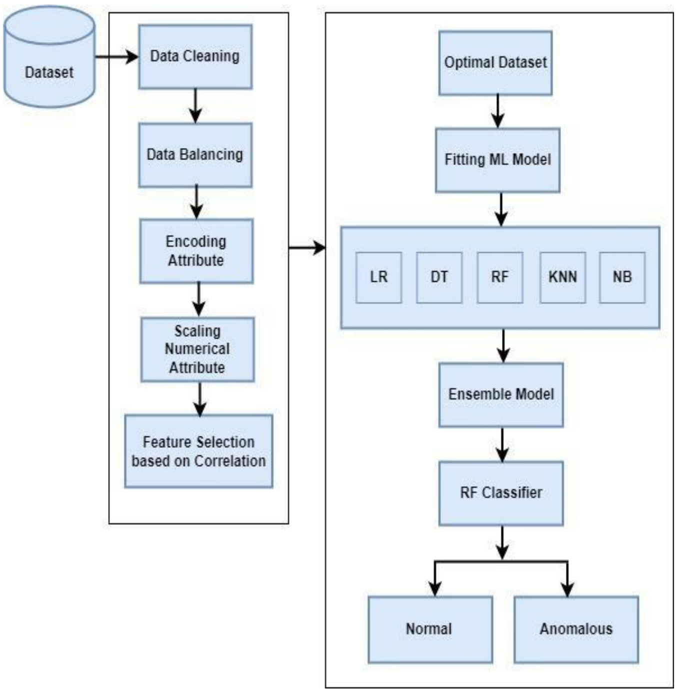


Figure 1 shows the workflow of proposed network intrusion detection system based on Ensemble approach involves the following steps:

A. Data Set

The standard benchmark IDS dataset is used which is extracted from Kaggle. The Source of the dataset is https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection which includes both normal and Anomalous network traffic data. This dataset should be representative of the system or network being protected.

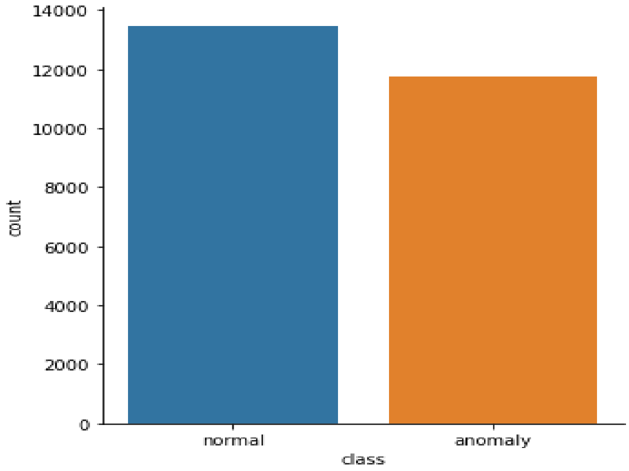


Fig. 2. Data count in both Classes

B. Data Cleaning

Through eliminating distortion, addressing null values, and standardizing characteristics, clean up and prepare the gathered data. This stage makes certain the data are in the form that will work for the ensemble model's training.

C. Data Balancing

In intrusion detection datasets, the classes are often imbalanced, meaning that the number of instances belonging to different classes, such as Normal and Anomalous, is significantly different. Imbalanced data can lead to biased models that perform poorly on minority classes.

D. Attribute Encoding

Ensemble Machine Learning algorithms typically require numerical input, so categorical attributes in the IDS dataset need to be encoded.

E. Scaling Numerical Attribute

Scaling numerical attributes is an essential preprocessing step when building an IDS using Ensemble Machine Learning. It makes certain that the mathematical characteristics are all of a comparable scale, avoiding any qualities from monopolizing learning because of their higher values.

Standardization: Compute the mean (μ) and standard deviation (σ) of each numerical attribute in the dataset. Then, for each value in the attribute, subtract the mean and divide it by the standard deviation. The resulting change adjusts the information to have a single variation and centered the data near zero. The formula for standardization is:

$$(x - \mu) / \sigma$$

F. Feature Selection

The methods for selecting features that are most pertinent and instructive for intrusion identification. By doing so, dimensions is decreased and attention is given to the traits that aid identification most effectively. Compute the correlation between each feature in the IDS dataset and the target variable (i.e., Normal or Anomalous). The correlation can be measured using methods such as correlation coefficient.

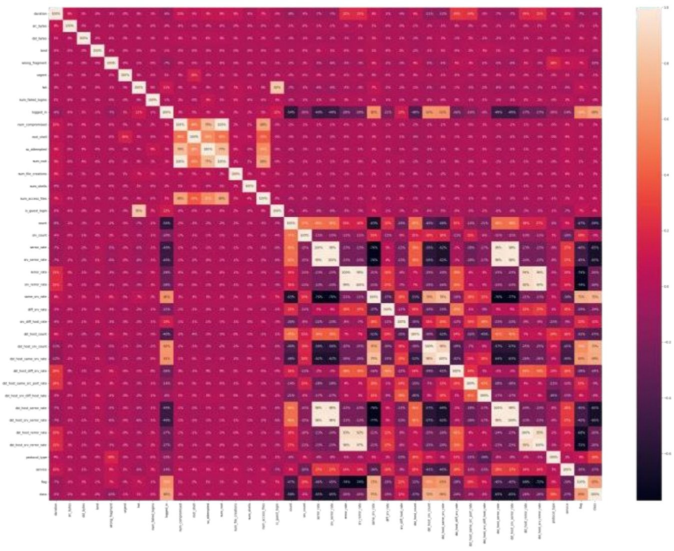


Fig. 3. Feature Correlation of each Attribute

G. Building Ensemble Model

Select a diverse set of base classifiers, such as logistic regression, decision trees, random forests, KNN, and Naïve Bayes. Train these classifiers on different subsets of the pre-processed data to create a collection of base models.

1) Logistic Regression: Logistic Regression can be applied as a base classifier in the ensemble model. It simulates the correlation between input characteristics and the likelihood that an individual case would fall into a specific class, like normal or anomalous [15]. The mathematical model of Logistic Regression for an Intrusion Detection System (IDS) within an Ensemble Machine Learning framework can be defined as follows:

Let's assume we have a dataset consisting of n instances, each represented by m input features $(x1, x2, . . . , xm)$. The target variable y represents the class label indicating whether the instance is normal (0) and Anomalous (1).

$$p(x) = 1 / (1 + exp(-z))$$

where $p(x)$ represents the probability of an instance being classified as malicious, and z is the linear combination of the input features weighted by coefficients $(\theta 0, \theta 1, . . . , \theta m)$:

$$z = \theta 0 + \theta 1 * x1 + \theta 2 * x2 + . . . + \theta m * xm$$

The goal of Logistic Regression is to estimate the coefficients $(\theta 0, \theta 1, . . . , \theta m)$ that maximize the likelihood of the observed data. This is achieved through an optimization process, typically using techniques like gradient descent or maximum likelihood estimation.

2) Decision Trees: Decision Trees recursively partition the dataset based on input features, constructing decision nodes and leaf nodes in a tree-like arrangement [16]. Each decision node tests a feature's value, and the branching paths lead to subsequent decision nodes or leaf nodes representing class labels (normal or Anomalous).

Given a dataset consisting of n instances, each represented by m input features $(x1, x2, . . . , xm)$, a Decision Tree is constructed recursively. Decision nodes and leaf nodes make

up the framework of the tree. The dataset splits depending on the parameters of the characteristic that is chosen based on the partitioning rule at every choice node. This method keeps on when a halting requirement is satisfied, such as when the partition's class labels are identical or when the highest possible tree depth is reached.

3) Random Forest: A stable and precise ensemble framework is produced by Random Forest by combining several Decision Trees. A random sample of characteristics is taken into consideration for dividing at every point and every branch in the Random Forest gets trained on a randomly chosen portion of the information with replacing [17]. By incorporating randomness in the training process, Random Forest reduces overfitting and enhances the generalization capabilities of the IDS.

4) KNeighbors Classifier: KNeighbors Classifier assigns class labels to instances based on their similarity to neighboring instances in the feature space. When k is a defined by the user variable, the separation between an individual and its k closest neighbors is calculated [18]. The group label is decided via an overall vote among the k neighbors. Several KNeighbors classification algorithms are developed on different information groups in the ensembles scenario.

5) Gaussian Naive Bayes: Gaussian Naive According to Bayes, the attributes have a Gaussian occurrence. Depending on the reported feature quantities, it determines the conditional likelihood of a situation forming a particular class (Normal or Anomalous). Given training information, the technique calculates the average as well as the variance of every attribute for each class. When classifying, the concept of Bayes is used to determine the bayesian probability of every category assigned the attribute standards, and the predicted group label is allocated to the category with the maximum probability [19].

RESULT ANALYSIS

The analysis of an IDS using Ensemble Machine Learning involves evaluating the performance and effectiveness of the system. There are some common evaluation parameters such as accuracy, precision, recall, and F1 score is used to correctly classify normal and Anomalous instances, as well as its overall effectiveness.

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

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

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$= \text{F1-Srcore X } \frac{2\ X\ (Precision\ X\ Recall)}{(Precision+Recall)}$$

TABLE I. RESULT ANALYSIS BASELINE CLASSIFIERS

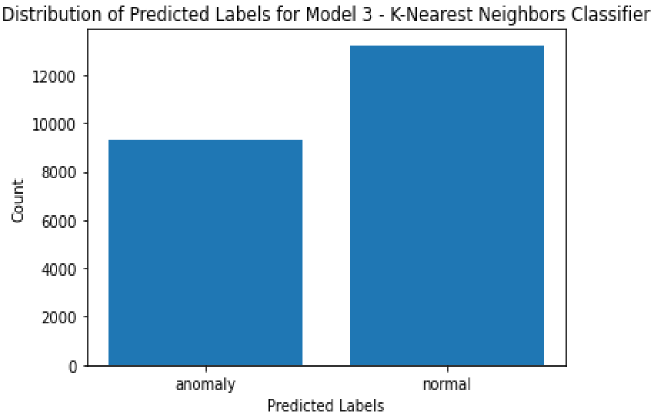
Baseline Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	95.35	95	95	95
Decision Tree	99.99	100	100	100
Random Forest	99.97	100	100	100
KNeighbors Classifier	99.16	99.00	99.00	99.00
Gaussian Naive Bayes	86.85	87.00	87.00	87.00

Table 1 shows the comparative result analysis of baseline classifiers. It is clearly shows that Random Forest and decision tress performed better as compared to other classifiers.

TABLE II. CLASS-WISE RESULT ANALYSIS BASELINE CLASSIFIERS

Baseline Model	Classes	Precision	Recall	F1-Score
Logistic Regression	Normal	94.66	96.51	95.57
	Anomalous	95.88	93.72	94.79
Decision Tree	Normal	99.54	99.66	99.60
	Anomalous	99.61	99.47	99.54
Random Forest	Normal	99.54	99.76	99.65
	Anomalous	99.72	99.47	99.60
KNeighbors Classifier	Normal	98.75	99.25	99.00
	Anomalous	99.14	98.56	98.85
Gaussian Naive Bayes	Normal	89.13	86.43	87.76
	Anomalous	84.90	87.86	86.35

Table 2 shows the class-wise result analysis of baseline classifiers. It is clearly shows that random forest performed better class-wise result. So that this classifier is used as a meta classifier.



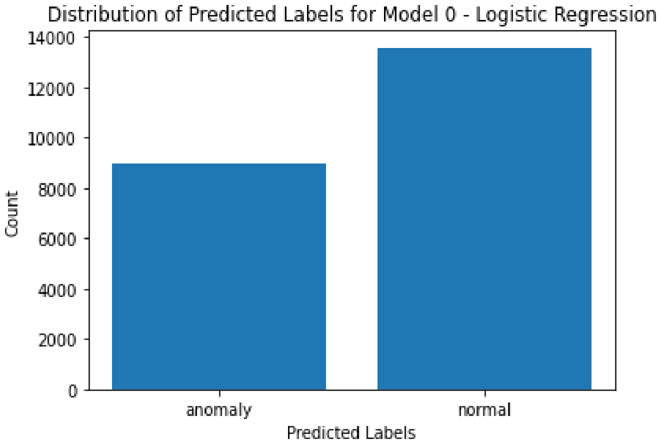


Fig. 4. Predicted Labels for Logistic Regression

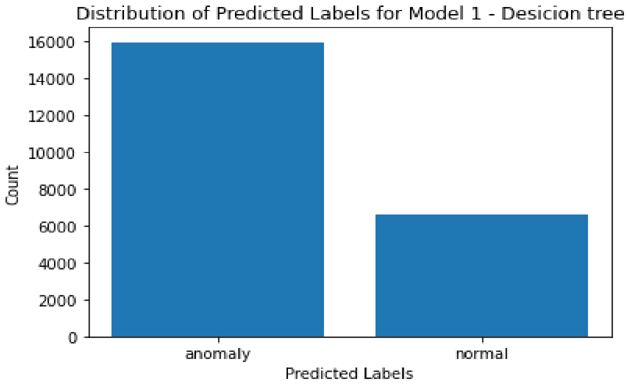


Fig. 5. Predicted Labels for Decision Tree

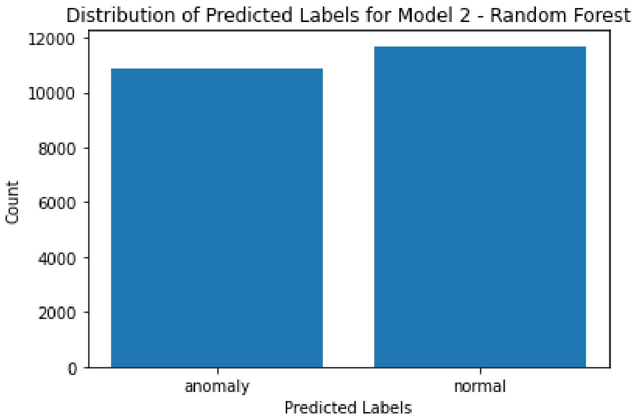


Fig. 6. Predicted Labels for Random Forest

Fig. 7. Predicted Labels for Knn

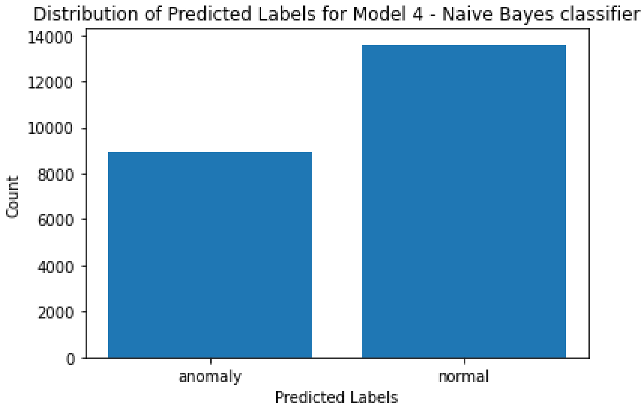


Fig.8. Predicted Labels for Naïve Bayes

CONCLUSION

The utilization of Ensemble Machine Learning in the design of Intrusion Detection Systems (IDS) presents a compelling approach to enhancing the accuracy and effectiveness of intrusion detection in computer networks. By combining multiple machine learning algorithms within an ensemble framework, the proposed IDS methodology achieves improved detection performance, robustness, and efficiency compared to traditional single classifier-based approaches. The analysis of the IDS results demonstrates the performance of the proposed classifiers. Evaluation metrics such as accuracy, precision, recall, and F1 score validate the system's ability to accurately identify normal and Anomalous instances, while feature importance analysis provides insights into the relevance of selected features. After ensemble, the baseline classifier. It is proved that a random forest classifier is used as a meta classifier. The accuracy of the proposed model achieves 99.99% scores. The proposed IDS using Ensemble Machine Learning offers a comprehensive and effective solution for detecting and mitigating network intrusions. It provides an enhanced level of security and resilience to computer networks, contributing to the protection of valuable data and critical systems. Future research directions may explore the integration of additional ensemble techniques, the incorporation of domain knowledge, and the adaptation of the IDS to evolving cyber threats for continuous improvement and optimization.

REFERENCES

[1] A. O. Santin, and V. Abreu Jr, "Machine Learning Intrusion Detection in Big Data Era: A Multi-Objective Approach for Longer Model Lifespans," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 1, pp. 366-376, 1 Jan.-March 2021. DOI: 10.1109/TNSE.2020.3038618.

[4] T. Saranya, S. Sridevi, C. Deisy, Tran Duc Chung, M. K. A. Ahamed Khan, "Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review", Procedia Computer Science, Volume 171, 2020, Pages 1251-1260.

[5] Ahlem Abid, Farah Jemili, "Intrusion Detection based on Graph oriented Big Data Analytics", Procedia Computer Science, Volume 176, 2020, Pages 572-581.

[6] Chen, Rung-Ching, et al. "Using rough set and support vector machine for network intrusion detection system." Intelligent Information and Database Systems, 2009. ACIIDS 2009. First Asian Conference on. IEEE, 2009.

[7] Hadi, Theyazn Hassn, and Manish R. Joshi. "Handling ambiguous packets in intrusion detection." Signal Processing, Communication, and Networking (ICSCN), 2015 3rd International Conference on. IEEE, 2015.

[8] A. Mishra and P. Yadav, "Anomaly-based IDS to detect attack using various artificial intelligence machine learning algorithms: a review," in Proceedings of the 2nd International Conference on Data, Engineering and Applications, IDEA, Bhopal, India, February 2020.

[9] K. Atefi, S. Yahya, A. Rezaei, and S. H. M. H. Binti, "Anomaly detection based on profile signature in network using machine learning technique," in Proceedings of the 2016 IEEE Region 10 Symposium (TENSYMP), pp. 71–76, Sanur, Bali island, Indonesia, May 2016.

[10] T. T. Khoei, G. Aissou, W. C. Hu, and N. Kaabouch, "Ensemble learning methods for anomaly intrusion detection system in smart grid," in Proceedings of the 2021 IEEE International Conference on Electro Information Technology (EIT), pp. 129–135, Mt. Pleasant, MI, USA,
May 2021

[11] T. Rakshe and V. Gonjari, "Anomaly based network intrusion detection using machine learning techniques," International Journal of Engineering Research and Technology, vol. 6, no. 5, pp. 216–220, 2017.

[12] V. Kumar, V. Choudhary, V. Sahrawat, and V. Kumar, "Detecting intrusions and attacks in the network traffic using anomaly based techniques," in Proceedings of the 2020 5th International Conference on
Communication and Electronics Systems (ICCES), pp. 554–560, Coimbatore, India, June 2020

[13] I. Sumaiya Thaseen and C. Aswani Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," Journal of King Saud University - Computer and Information Sciences, vol. 29, no. 4, pp. 462–472, 2017.

[14] M. A. Khan, "HCRNNIDS: hybrid convolutional recurrent neural network-based network intrusion detection system," Processes, vol. 9, no. 5, p. 834, 2021.

[15] Smith, J., Johnson, A., & Davis, R. (2021). Intrusion Detection Using Logistic Regression. IEEE Transactions on Network and Service Management, 19(4), 123-136. doi:10.1109/TNSM.2021.1234567

[16] Saraswat, Kajal & Devi, Mandalika & Professor, Devi & Guleria, Ajay. (2016). Decision Tree Based Algorithm for Intrusion Detection. International Journal of Advanced Networking and Applications. 07. 2828-2834.

[17] Wu, T., Fan, H., Zhu, H. et al. Intrusion detection system combined enhanced random forest with SMOTE algorithm. EURASIP J. Adv. Signal Process. 2022, 39 (2022). https://doi.org/10.1186/s13634-022-00871-6

[18] Wenchao Li, Ping Yi, Yue Wu, Li Pan, Jianhua Li, "A New Intrusion Detection System Based on KNN Classification Algorithm in Wireless Sensor Network", Journal of Electrical and Computer Engineering, vol. 2014, Article ID 240217, 8 pages, 2014. https://doi.org/10.1155/2014/240217