CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
https://cajmtcs.centralasianstudies.org/index.php/CAJMTCS
Volume: 05 Issue: 04 | October 2024 ISSN: 2660-5309

Article

A Comparative Analysis of an Intrusion Detection System in Network Architecture Based on Machine Learning Technology

Inas Saadi Abed[1]*

1. College of Education for Pure Science, University of Diyala, Diyala, Iraq
* Correspondence: Inassaadi1984@gmail.com

Abstract: Network infrastructure is kept safe from unauthorized entry and phishing by intrusion detection systems (IDS). Traditional IDS solutions are often unable to detect tough cyber threats properly and, at the same time, reduce false positives due to an ever-increasing number of permutations. In the following paper, we introduce an IDS framework that carries on real-time decision-making based on machine learning and feature selection mechanisms. Hence, we will be classifying the machine learning classifiers like random forests, support vector machines, and decision trees on a common collection. Reducing the computational complexity & improving on model accuracy and feature space optimization by RFE followed with PCA. Our experiments show that the use of machine learning and feature selection to be embedded into IDSs makes these forthcoming systems more effective compared to traditional ways with high true positive proportions combined with low false positives. In this paper, we have proposed a broad perspective on how machine learning and feature selection help to grow the new and most effectively powered IDS as per the secured solution.

Keywords: Intrusion Detection, Machine Learning, Infrastructure of Network, Decision-making, Classifier

Citation: Abed, I, S. A Comparative Analysis of an Intrusion Detection System in Network Architecture Based on Machine Learning Technology. Central Asian Journal of Mathematical Theory and Computer Sciences 2024, 5(4), 456-464.

Received: 10th July 2024
Revised: 11th August 2024
Accepted: 24th Sept 2024
Published: 27th Oct 2024

1. Introduction

Current trends in technology have made network security vital for individuals as well as organizations to enquire about [1][2]. With network attacks on the rise both in frequency and complexity, there has never been a more important time to see to it that we have advanced security mechanisms like these protecting our infrastructures. Intrusion detection systems (IDS) are important parts of network security paradigms that exist to detect and analyze traffic for real-time alerts related to the presence of unwanted packets within a packet stream or trust violation, which can occur when communicating between trusted devices. Previous IDSs have relied on signature-based detection methods, which are based on known malicious behavior patterns. Despite this, it is challenging for these signature-based systems to cope with novel and emerging attack patterns, which can result in an increase in false positives or no detections [3]. The introduction of machine learning (ML) has completely reshaped intrusion detection. It is a much smarter, more sophisticated, and more versatile technology for detecting threats or anomalies. An ML-based IDS can analyze a lot of historical data so that it can identify complex scenarios of network activity patterns. What is more, such systems possess the unique feature of

recovering from new and previously unknown intrusion types without obeying strict, predefined rules, which, as a rule, are part of traditional detection systems. The latest development holds great promise to substantially enhance the accuracy and efficiency of intrusion detection operations, which makes ML a trusted development of IDS that is growing to meet the increased complexity of cyber attackers. It is safe to say that the integration of machine learning is the new frontier for the future of network security as IDS matures [4, 5].

The focus of this research is the evaluation and testing for comparison purposes of intrusion detection systems in network architectures using machine learning algorithms [6]. Several ML models have been examined in this study, such as support vector machines, decision trees, or random forests, in order to evaluate their viability in terms of the three-defining metrics: scalability, speed, and accuracy of detection. In addition, the paper investigates whether feature selection and dimensionality reduction approaches can be used to improve the process of intrusion detection and cost-effectiveness in using such models. The study intends to analyze several approaches to demonstrate the capabilities and shortcomings of the selected strategies and provide recommendations on the most optimal methods for improving security networks in the context of increasing threats to the cyber world by optimizing machine learning-based intrusion detection systems in this context [7].

The rest of the paper is organized as follows. Relevant work is explained in Section 2. In Section 3, the preliminaries and recommended approaches are explained. The application of the suggested strategy is in Section 5, which is where all of the results are displayed. Section 6 provides the conclusion and future work.

Related Work

In recent years, the utilization of machine learning (ML) in intrusion detection systems (IDS) has garnered substantial attention as a result of its capacity to enhance the detection of intricate and evolving cyber threats. In order to overcome the constraints of conventional IDS, researchers have explored a variety of ML techniques, including the reduction of false positives, the adaptability to new attack types, and the accuracy of detection. This section evaluates the current state of IDS research, which is primarily based on machine learning technology. It also identifies gaps that this study intends to address and emphasizes critical contributions. Tavallaee et al. [8] and Farid et al. [9] have recently applied machine learning algorithms to well-known datasets such as KDD Cup and NSL-KDD. These studies have demonstrated the potential of ML models, including Decision Trees, SVM, and Naive Bayes, to achieve higher detection rates than signature-based systems. Kumar et al. [10]. Their findings indicated that ensemble methods, including Random Forests, typically outperform individual classifiers in terms of both robustness and detection accuracy. In the same vein, Shafiq et al. [11] demonstrated that the detection of sophisticated attacks, including Distributed Denial of Service (DDoS) and Advanced Persistent Threats (APT -k), can be enhanced through the use of deep learning performances, particularly Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Nevertheless, the computational complexity of deep learning models can be a constraint, particularly in real-time applications.

In conjunction with machine learning algorithms, feature selection has emerged as a critical element in the optimization of IDS performance. The computational overhead can be considerably reduced while maintaining or improving detection accuracy through the use of dimensionality reduction performances, such as principal component analysis (PCA) and recursive feature elimination (RFE), as demonstrated by Zaman et al. [12]. These techniques enable the intrusion detection system (IDS) to concentrate on the most pertinent aspects of the information, thereby attractive the efficacy and speed of the interruption exposure process.

The development of robust IDS systems continues to be a significant challenge, despite the progress that has been made. Scalability across large networks, real-time detection, and the capacity to manage asymmetrical data sets (in which normal traffic significantly exceeds malicious activity) are perpetual obstacles. Ahmed et al. [13] utilized synthetic data generation techniques, including the Synthetic Minority Over-sampling Technique (SMOTE), to enhance the detection rate of minority assault classes, thereby resolving the issue of class imbalance. Therefore, additional research is required to achieve a balance between real-time performance and high detection accuracy, particularly in the context of large networks.

This paper expands upon the aforementioned work and performs a comparative analysis of a variety of ML performances, such as traditional algorithms and ensemble methods, to assess their efficacy in network-based IDS. Furthermore, in directive to increase the recital of the typical, we implement feature selection methods that take into account both computational efficiency and detection accuracy. This study contributes to the ongoing endeavors to create IDS systems that are more efficient, scalable, and accurate by concentrating on these critical aspects, which are based on machine learning technologies.

2. Materials and Methods

Preliminaries

a. ML Classifier

One type of gadget that assigns previously undetected patterns to the relevant categories is known as a classifier. In the beginning, the classifier is given something called training information, which it uses to construct a decision model. The model is then given some data that it has not seen before. Examples of classifications to consider. In this research article, the following is a condensed description of some of the classification approaches that are employed the most frequently:

i. Support Vector Machine: This algorithm is founded on the principle of structural risk minimization, which provides it with an advantage in terms of scalability and performance [14]. In a multidimensional space, the fundamental concept of a SVM is to identify a decision boundary that distinguishes between unobserved patterns in distinct classes.

ii. Decision Tree (DT): DT algorithms are a well-known instrument for arrangement and estimate tasks. They construct a model that forecasts the yield of a pattern by analyzing the various input attribute values of the pattern. No domain knowledge or parameter configuration is necessary for the construction of DT; rather, the provided data set is learned and modeled. DT is composed of three fundamental components: the conclusion protuberance, the edges or divisions, and the leaf node [15].

iii. k-nearest neighbor algorithm (k-NN) is the most straightforward of all machine learning algorithms. The output is determined by the k closest neighbors or k training patterns [16]. The output scheming is contingent upon the completion of the mission. For instance, when a pattern is classified as unknown, the pattern is designated as the class that seems most regularly between the nearest drill patterns.

iv. Random Forests (RF) are learning algorithms that combine multiple decision trees to improve the accuracy of classification or regression. Each tree is trained on a random subset of the data, and the features make predictions independently. The final result is based on a majority vote for classification or regression averaging. This approach reduces the risk of overfitting and increases power, as errors in individual trees are

reduced by aggregation. Random forests are highly effective in processing large data sets and complex interactions between features, making them popular for tasks such as fraud detection, medical diagnostics, and more [17].

b. Feature Selection (FS)

In the process of feature selection, the procedure of choosing the subset of the attribute is performed from the entire set. The learning algorithm takes less time to run, as this set of representatives only has relevant and important characteristics. Generalize and learn a coarser classifier that comes out when unnecessary and extraneous features are stripped from the set. Further, feature selection is useful in data visualization as well as data comprehension and understanding. A brief description of some of the most common feature selection performances used in this paper is given below [18].

i. The most used technique in dimensionality reduction to extract as many variables as possible from big data is principal component analysis. Liner combines the initial features as principal components. This process yields a new set of variables called the principal components. The figures show that the first few PCs exhibit a high degree of linearity. The next component is perpendicular to the previous component, and it is noted that those components are the high variance components because they capture most of the data. One advantage it brings is that it enables reduction of the number of features, which enhances computation time, and in situations where there is high dimension and or redundancy, the performance of the model is improved.

ii. When it comes to model training, Recursive Feature Elimination, also known as RFE, is a feature selection technique that is used to help decide which characteristic is most important. This is made possible through the repeated creation of a model and removal of the least relevant features, one after the other, based on the relevance scores of those features. Further, it is necessary to proceed with this idea as long as the required amount of characteristics is not found. RFE is often applied when used in combination with other machine learning classifiers like SVM and Random Forests, among others. This falls into place because RFE can increase the model performance without distorting the generalization and can reduce over-fitting to the most significant features.

3. Results

In the experimentation stage of the present research, the performance of each of the different ML algorithms to engage in in a network architecture to detect intrusions was evaluated. The feature selection process yielded DT, RF, SVM and k-NN as the classifiers to be used. For the same purpose, we also use feature selection methods such as PCA and RFE to get better sensitivities and specificities and reduce the computational overhead. The findings of these analyses are presented here, excluding the time measurement, F1-measure, recall, and detection performance for maximum effectiveness.

Dataset and Experimental Setup

The NSL-KDD dataset was used to carry out the experiments due to its reputation as a benchmark for network systems. This dataset contains pre-labelled data related to different types of network traffic and their activities, including common and malicious activities such as DoS, probes, and R2L attacks. Eighty per cent of the data was used as training data, while the remaining 20% was used as testing data. Indeed, all the classifiers

were trained and tested on the same data split and preprocessed through the standard normalization methods so that the results could be comparable. Two situations were examined for each algorithm: one with all features and the other using only selected features, where we used P, CA and Recursive Feature Elimination (RFE).

Accuracy

Accuracy of detection was the most common criterion employed when comparing one classifier to another. Overall, the random forest outperformed other algorithms, with no feature selection indexing 96.8%, as indicated in Table 1 and Figure 1. The second-best performance resulted from SVM at 94.4% and then DT, which maintained 90.2%, while k-NN at a comparatively low 88.7%. Applying feature selection methods also enhanced the performance of both SVM and K-NN classifiers, particularly by using the PCA technique. This process decreases the dimensionality of data and emphasizes the most important features that, in turn, can be easily classified with the help of classifiers distinguishing between data classes. When evaluating the system on the basis of PCA, the rate of SVM was raised to 95.4%, and the accuracy of K-NN was raised to 91.2%. These enhancements reveal that PCA can be extraordinarily valuable in measures that depend on the size of feature space as it can assist the algorithms in performing the classification job effectively.

Meanwhile, the latter, a more complicated ensemble learning technique, discovered only a slight accuracy of 97.1% when feature selection was employed. This result implies that random forests do not largely rely on feature selection for the best performance since they are capable of processing large numbers of features at once and performing selection through random samples and fusion. Therefore, the stable results of random forests, in this case, prove its variability and effectiveness in different databases, and thus, classification tasks without feature selection are also highly recommended.

Table 1. Accuracy Performance of different classifiers with FS

Algorithm	Accuracy (No FS)	Accuracy (PCA)	Accuracy (RFE)
DT	90.2%	92.5%	92.0%
RF	96.8%	97.1%	97.0%
SVM	94.4%	95.4%	95.2%
k-NN	88.7%	91.2%	90.6%

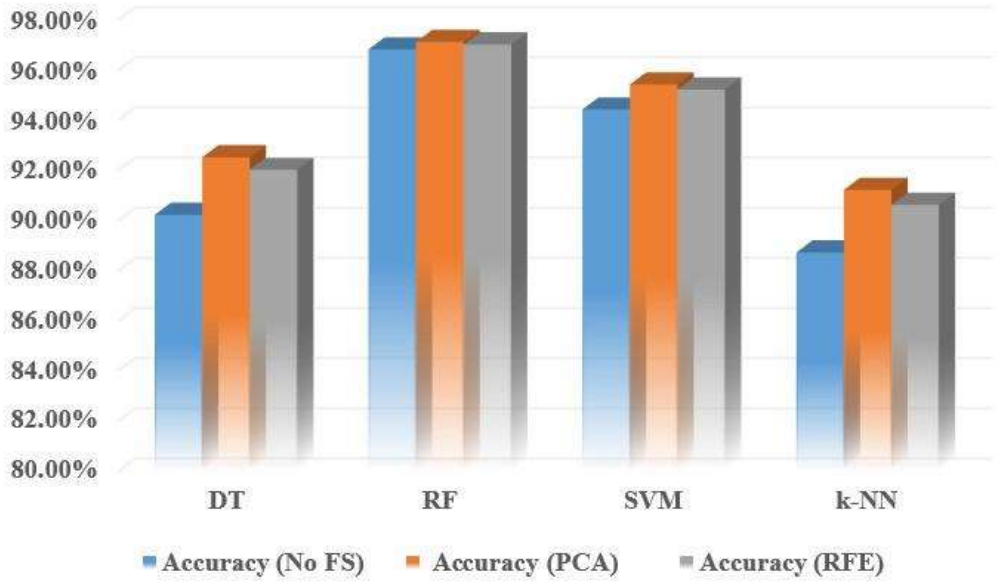

Figure 1. The comparison of the accuracy performance on various classifiers in context to FS

Precision, Recall, and F1-Score

To gain a deeper empathy for classifier performance, we measured additional metrics beyond accuracy: accuracy and error, as well as point measurements such as precision, recall, and its equivalent of the f1 score. This measure gives information on how each classifier is efficient in classifying normal and 'abnormal' or 'anomalous' data, especially the Weiss' categories of basic, DoS, probe, User2Root (u2r) and Remote2Root (r2l) attack types. These attacks are generally of lower frequency and are less salient in the dataset, and thus have a lower probability of detection by the classifiers. Table 2 and Figure 2 show the mean of average precision, recall, and f1 scores for all attack classes with a broad view of all algorithms' performance. In all these aspects, the RF is still better with precision at 95.6%, recall at 96.2% and f1 score at 95.91%. This high value shows that the random forest algorithm not only detects the attacks but also has low false positive and false negative rates for both frequent attacks and special attacks. This means that SVM also exhibited comparable accuracy and had an excellent f1 score of 94.8%, thus proving that this model could easily identify both the frequent and the rare attack patterns. The nearly equipercentile distribution of accuracy and memory in SVM shows the capacity of the classifier for response to different types of attacks as a basic solution for comprehensive cybersecurity tasks with high precision for all kinds of attacks. K-NN and decision tree implementations are less favorable in this case, particularly in cases of less frequency u2r and r2l attacks. Of this, there is a strong discrepancy at lower recall values, which means that this classifier is more likely to fail to identify these attacks. That is why lower precision, recall, and f1 scores for K-NN and decision trees can be stated: It is less applicable when precision has to be maintained as equally frequent and infrequent attack classes to be performed constantly. Therefore, their performance may be more variable, depending on the specific distribution of attack types in the data.

Table 2. Performance of classifiers of other metrics

Algorithm	Precision	Recall	F1-Score
DT	88.9%	89.3%	89.1%
RF	95.6%	96.2%	95.91%
SVM	94.5%	95.1%	94.8%
k-NN	87.1%	88.4%	87.7%

Computational Effectiveness

The computational performance is evaluated based on the training time, prediction time, and overall resource utilization during the training and testing phases. Random forests, despite having superior accuracy, exhibit relatively high computational costs due to the collaborative countryside of the algorithm. The training time for the random forest is, on average, 10.4 seconds, much higher than the DT (3.6 seconds) and k-NN (4.3 seconds). SVMs exhibit modest training times, on average 6.9 seconds, but are computationally intensive when working with larger feature sets. However, with feature selection techniques such as PCA, the computational load for SVM and k-NN was significantly reduced, reducing the training time by almost 30%.

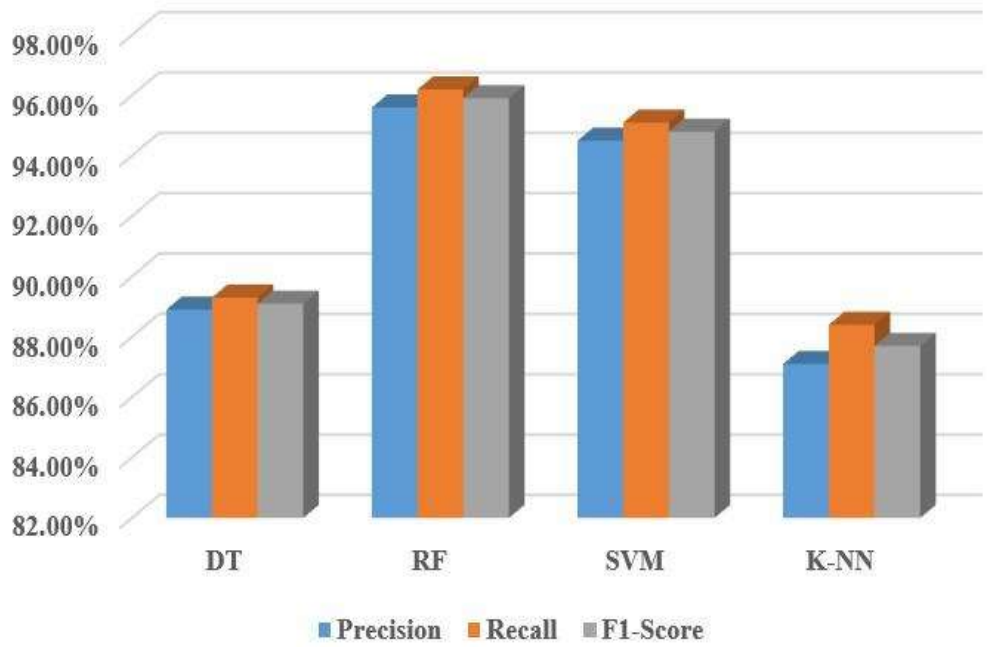

Figure 2. Comparison of the performance of various classifiers

RF shows minimal improvement in training time with feature selection, indicating that it cannot benefit as much from dimension reduction as other classifiers.

Table 3. Computational Effectiveness

Algorithm	Training Time (No FS)	Training Time (PCA)	Training Time (RFE)
DT	3.6s	3.2s	3.3s
RF	10.4s	9.8s	9.9s
SVM	6.9s	4.7s	4.9s
k-NN	4.3s	2.9s	3.1s

4. Discussion

Based on the findings of the experiments, it has been determined that RF, despite having a greater computational cost, provide the uppermost overall performance in terms of detection accuracy, precision, and recall. SVM also performs well, particularly when paired with feature selection techniques such as PCA. Such a combination strikes a balance between the accuracy of detection and the efficiency of computation. Even if they are successful, k-nearest neighbors and decision trees are better suited for situations in which computational resources are restricted, and a somewhat lower level of precision is acceptable and acceptable. The selection of features is an essential component in increasing the routine of most ML models. This is accomplished by decreasing the dimensionality of the data, which in turn improves both accuracy and efficiency. SVM and k-NN, on the other hand, are the algorithms that profit the most from feature reduction. However, the level of improvement varies from method to algorithm [19].

These findings provide significant insights into the application of ML for network intrusion detection. They also demonstrate that selecting the appropriate algorithm and optimizing feature sets are essential for attaining a balance among accuracy and productivity in intrusion detection systems [20].

5. Conclusion

This paper proposes an IDS model that compares the performance of different combinations. A feature selection algorithm selects a subset of important features and then uses the important set feature to train various types of classifiers. RFE and PCA selection techniques and k-NN, DT, and SVM classifiers are used in this paper. The paper uses different feature selection algorithms and classifiers because each classification and feature selection algorithm have many advantages and disadvantages. It is difficult to choose one or the other to implement an intrusion detection system. In addition, the experimental results show that machine learning can be used in intrusion detection, as all combinations provide considerable accuracy. SVM and DT classifiers perform better than others, and among feature selection methods, the RFE feature selection method is better than others. The highest accuracy in all combinations is for the selection of RFE elements with DT. Thus, it can be concluded from this study that the combination of RFE and DT feature selection can be used to design effective intrusion detection. To ensure data security, we may implement a wide variety of feature selection methods in future studies.

REFERENCES

1. Ahmed, M., Mahmood, A. N., & Hu, J. (2021). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19-31. https://doi.org/10.1016/j.jnca.2015.11.016
2. Farid, D. M., Harbi, N., & Rahman, C. M. (2013). Combining Naive Bayes and Decision Tree for Adaptive Intrusion Detection. International Journal of Network Security & Its Applications, 5(2), 1-11. https://doi.org/10.5121/ijnsa.2013.5201
3. H. K. Mistry, C. Mavani, A. Goswami, and R. Patel, "A Survey Visualization Systems for Network Security," *Educational Administration: Theory and Practice,* vol. 30, no. 7, pp. 805-812, 2024.
4. Heidari and M. A. Jabraeil Jamali, "Internet of Things intrusion detection systems: a comprehensive review and future directions," Cluster Computing, vol. 26, no. 6, pp. 3753-3780, 2023.
5. Huang, R. Xu, Y. Chen, and M. Guo, "Research on multi-label user classification of social media based on ML-KNN algorithm," Technological Forecasting and Social Change, vol. 188, p. 122271, 2023.
6. J. Andrade-Hoz, J. M. Alcaraz-Calero and Q. Wang, "Multi-Layer Multi-Technology Firewall Optimisation in Beyond 5G Networks Using Machine Learning Classifiers," *2024 14th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, Rome, Italy, 2024, pp. 563-568, doi: 10.1109/CSNDSP60683.2024.10636539.
7. J. Banerjee, S. Maiti, S. Chakraborty, S. Dutta, A. Chakraborty, and J. S. Banerjee, "Impact of machine learning in various network security applications," in 2019 3rd International conference on computing methodologies and communication (ICCMC), 2019, pp. 276-281: IEEE.
8. Kumar, S., Kalia, S., & Mittal, M. (2019). A Comparative Study of Machine Learning Algorithms for Network Intrusion Detection. In 2019 International Conference on Computing, Communication, and Automation (pp. 1-5). IEEE. https://doi.org/10.1109/ICCCA.2019.00009
9. L. H. Jyothula, G. Eppa, A. V. Indhukuri, and M. J. Mehdi, "Lung Cancer Detection in CT Scans Employing Image Processing Techniques and Classification by Decision Tree (DT) and K-Nearest Neighbor (KNN)," in 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN), 2023, pp. 1-5: IEEE.
10. P. Dini, A. Elhanashi, A. Begni, S. Saponara, Q. Zheng, and K. Gasmi, "Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity," Applied Sciences, vol. 13, no. 13, p. 7507, 2023.
11. P. Palsodkar. "Applied Nonlinear Analysis, Machine Learning, and the Opening of New Realms in Virtual Communication Technology". *Advences in Nonlinier Variational Inequelities 27 (1),* pp. 1-17.
12. R. Das and T. H. Morris, "Machine learning and cyber security," in 2017 international conference on computer, electrical & communication engineering (ICCECE), 2017, pp. 1-7: IEEE.
13. S. Hermiati, R. Herteno, F. Indriani, T. H. Saragih, and T. Triwiyanto, "A Comparative Study: Application of Principal Component Analysis and Recursive Feature Elimination in Machine Learning for Stroke Prediction," Journal of Electronics, Electromedical Engineering, and Medical Informatics, vol. 6, no. 3, pp. 231-242, 2024.

14. S. S. Vellela et al., "Improving Network Security Using Intelligent Ensemble Techniques: An Integrated System for Detecting and Managing Intrusions in Computer Networks," in 2024 International Conference on Advances in Modern Age Technologies for Health and Engineering Science (AMATHE), 2024, pp. 1-7: IEEE.
15. S. Suthaharan and S. Suthaharan, "Support vector machine," Machine learning models and algorithms for big data classification: thinking with examples for effective learning, pp. 207-235, 2016.
16. Shafiq, M., Tian, Z., Sun, Y., Du, X., & Guizani, M. (2020). Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city. Future Generation Computer Systems, 107, 433-442. https://doi.org/10.1016/j.future.2020.02.017
17. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD Cup 99 dataset. In 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (pp. 1-6). IEEE. https://doi.org/10.1109/CISDA.2009.5356528
18. Y. Guangxu, "Research on computer network information security based on improved machine learning," Journal of Intelligent & Fuzzy Systems, vol. 40, no. 4, pp. 6889-6900, 2021.
19. Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, and X. Liang, "An improved random forest based on the classification accuracy and correlation measurement of decision trees," Expert Systems with Applications, vol. 237, p. 121549, 2024.
20. Zaman, N., Karray, F., & Alemzadeh, H. (2016). Feature selection using principal component analysis for optimal Fuzzy c-Means clustering on health data. IEEE Transactions on Fuzzy Systems, 24(1), 121-133. https://doi.org/10.1109/TFUZZ.2015.2438517.