Machine Learning-Based Network Intrusion Detection System

Sumedha Seniaray and Rajni Jindal

Abstract As the network is dramatically extended, security has become a significant issue. Various attacks like DoS, R2L, U2R are significantly increasing to affect these networks. Thus, detecting such intrusions or attacks is a major concern. Intrusions are the activities that breach the system's security policy. The paper's objective is to detect malicious network traffic using machine learning techniques by developing an intrusion detection system in order to provide a more secure network. This paper intends to highlight the performance comparison of various machine learning algorithms like SVM, K-Means Clustering, KNN, Decision tree, Logistic Regression, and Random Forest for the detection of malicious attacks based on their detection accuracies and precision score. A detailed analysis of the network traffic features and the experimental results reveal that Logistic Regression provides the most accurate results.

Keywords Intrusion · Intrusion detection · Network-based intrusion detection system · Network security · Machine learning · Network traffic · KDD · Feature selection

1 Introduction

An attack is any kind of action that threatens the integrity, confidentiality; attempting to achieve unauthorized access to the sensitive information of a network system is known as an attack. An Intrusion Detection System is a system that helps detect a variety of malicious or abnormal network traffic and computer usage that is not feasible to detect with the help of a conventional firewall or is unknown to the user. This comprises of network attacks against all the services that are vulnerable, data-driven attacks on applications, host-based attacks like unauthorized system or

S. Seniaray · R. Jindal (✉)
Delhi Technological University, New Delhi, India
e-mail: rajnijindal@dce.ac.in

S. Seniaray
e-mail: sumedhaseniaray@dtu.ac.in

S. Smys et al. (eds.), *Computer Networks and Inventive Communication Technologies*,
Lecture Notes on Data Engineering and Communications Technologies 75,
https://doi.org/10.1007/978-981-16-3728-5_13

software login, privilege escalation, and access to personal/sensitive user files and data, and malware (viruses, worms, and trojan horses). Intrusion detection systems and firewalls both are a part of network security, but they differ from each other as firewall looks on the outside for intrusions so that intrusions can be stopped before happening. Firewalls forbid access between networks so that intrusion can be prevented. If an attack is within the network, then it does not signal. In contrast, an Intrusion Detection System (IDS) detects a suspicious intrusion once it has occurred and then signals an alarm to notify that an intrusion has been detected. Firewalls are like barriers that protect the system from the outside threats and signals the system if unauthorized or forceful attempts are made from the outside. In contrast, an Intrusion Detection System signals the system when it detects such malicious activity. The main agenda of IDS is to protect the host or the network from any malicious or unusual activity that can enter the system and compromise the data. Thus, the aim is to detect an intrusion before the hackers get to the information and damages or corrupt it. Intrusion detection can be performed for various application areas, for instance, in Digital forensics, in IoT [1] for detection of intrusions in the network, wireless sensor networks (WSN) [2], social media networks [3], real-time security systems, and also in combination with firewalls for additional security of the network as well as the host system (Fig. 1).

Intrusion Detection Systems are of two types:

- **Network-based Intrusion Detection System (NIDS)**

NIDS [4] detects any threat or intrusion like denial of service (DoS), etc., introduced in the network by keeping track of the network traffic. A network-based intrusion detection system resides on the network monitoring the network traffic flows, that is, the inbound and outbound traffic to and fro from all the devices connected in the network.

- **Host-based Intrusion Detection System (HIDS)**

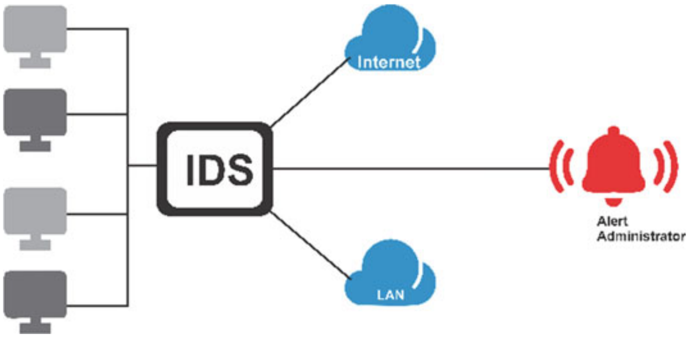

Fig. 1 Intrusion detection system

HIDS is installed on the client's system and helps detect any threats which are introduced in a specific host via the network packet is received, or which host accessed the data or whether any unauthorized access has been done, etc.

On the basis of methods of detection, Intrusion Detection System (IDS) can be categorized as:

- **Signature-based IDS**

Signature-based IDS [5] provides significant intrusion detection results for well-known, specified attacks. Thus, they are not very capable of detecting unfamiliar or new attacks.

- **Anomaly-based IDS**

An anomaly from the perspective of security is an event that is suspicious. Anomaly-based IDS [4, 6] have the potential to detect previously unknown, unseen intrusion events or attacks. Therefore, anomaly-based IDS can detect both known and unknown intrusions. Thus anomaly-based systems have a higher rate of detection capability than signature-based IDS.

We aim to work on the anomaly-based network intrusion detection system. Thus, monitoring the network traffic flows in order to detect not just the known but also unknown or abnormal network traffic flow. An anomaly-based intrusion detection system monitors the network traffic and compares it to the normal traffic flows, and if it detects some unusual pattern or anomalies on the network, it alarms the signal indicating a potential threat. Based on this comparison, the network traffic flow is categorized as “normal” or “abnormal or malicious".

There are various Machine Learning techniques that are incorporated into the intrusion detection procedure to decrease the false alarm rates. Machine Learning is also used to automate the building of an analytical model. As Machine Learning is a part of Artificial Intelligence, which prevails on the concept that a system gets trained, makes decisions, and learns to diagnose patterns with fewer human intervention; thus, it is determined to build a model that enhances its performance on the basis of previous results. For this purpose, various Machine Learning techniques are Support Vector Machine (SVM), KNN, Random Forest, Logistic Regression, Decision tree, and Naïve Bayes, etc.

To give a detailed analysis of these algorithms for detection of intrusion and to establish our anomaly-based network intrusion detection system, we collected the non-malicious network traffic, that is, KDD’99 data set and malicious network traffic, to train these machine learning classifiers on the basis of the collected network traffic data. The main contributions of this paper are summarized as follows:

1. Collection of both non-malicious or normal data and malicious data.
2. After analyzing the network traffic, feature extraction is performed, and 14 traffic features are extracted.
3. Feature selection is performed using the Feature Importance technique on these 14 extracted features to inherit the most significant ones, on which Machine Learning techniques are performed.

4. Machine learning classifiers are then trained on individual features to identify the intrusion detection accuracy and precision.
5. A combination of network traffic features is done, which lie above 50% on the Feature Importance Scale.

2 Related Work

In this modern era of Machine Learning, network intrusion detection system has become a vital component in network security. In today's period, it is vital for the organization and individual to secure their computer and network data as once the network is compromised, it can cause a lot of information damage. Various machine learning algorithms are applied to intrusion detection systems, such as decision tree [7–10], Logistic Regression [11–13], Support Vector Machine (SVM) [14–16], and Random Forest [15, 17]. In [17], a Random Forest-based intrusion detection system model was developed where the effectiveness of the Random Forest-based intrusion detection system model was tested on the NSL-KDD dataset, and it was noticed that the Random Forest performance was slow for real-time predictions when the number of trees was increased. Their results exhibited a detection rate of 99.67% in comparison with J48. In [15], detailed analysis and comparison are drawn on different machine learning algorithms, namely Random Forest, Support Vector Machine (SVM), and Extreme Learning Machine (EML), to find out the algorithms which give a better result when to amount of data to be analyzed is increased. And it was realized that Extreme Learning Machine (EML) gives the best results when the entire data is taken into consideration. When half of the data was considered, SVM performs better than the other two. In [16], the overall performance of SVM is improved by accelerating the convergence of the algorithm and increasing its training speed. A new function was created with the intention that the error rate of the SVM is reduced. Repalle and Kolluru [18] discovered that it is crucial to obtain a well-labelled dataset in order to provide efficient results. K-Nearest Neighbour (KNN) was found to be the best working algorithm; for the analysis, the values assigned to the variable 'K' is of importance. Fayyad et al. [19] discusses a comprehensive analysis of cybersecurity with the help of intrusion detection using machine learning (ML) and data mining (DM) methods, where performances of both ML and DM techniques are addressed to analyze accuracies of each of these techniques, which contributes to the field of cybersecurity. Tao et al. [15] proposed the FWP-SVM-GA algorithm, an intrusion detection algorithm that is based on the characteristics of the Support Vector Machine (SVM) and Genetic algorithm (GA) algorithm where the FWP-SVM-GA algorithm performs feature selection, parameter optimization of SVM based on GA. This reduces the SVM error rate and enhances the true positive rate. Finally, an optimal feature subset is used on the feature weights and SVM parameters in order to optimize them. As a result, classification time, error rates decrease, and the true positive rate increases. According to [20], intrusion detection is considered as a multi-class and two-class classification. This is performed using the SVM machine learning algorithm. SVM acts as a decision-making model throughout the training phase in the proposed SVM-based intrusion detection model. They performed three kinds of experiments on the 1999 KDD dataset, wherein they performed the experiments on 41 features set and presented a comparison of SVM IDS with KDD 1992 contest winner and concluded that SVM IDS is more effective when it comes to intrusion detection. In [21], an intrusion detection framework using SVM along with feature augmentation is performed on NSL-KDD dataset. Feature augmentation was done in order to provide a more concise and high-quality training data set for the SVM classifier, which helped improve the efficiency of the SVM-based proposed model. As a result of the experiment, the proposed model achieved a high detection accuracy of 99.18%.

3 Methodology and Implementation

This section describes the way the machine learning classifiers are implemented on the network traffic features to design an anomaly-based intrusion detection model. The implementation is summarized in four phases, represented in Fig. 2, namely: (1) Network traffic collection (2) Data pre-processing (3) Feature Extraction and Feature Selection (4) Implementation of Machine Learning (ML) techniques for the proposed intrusion detection system.

3.1 Network Traffic Collection

As the estimation and analysis of the machine learning techniques are performed on the network traffic, we need two sets of network traffic data, that is, malicious or intrusive traffic data and non-malicious or normal traffic data. The normal traffic data via an extensive network traffic analyzer software Wireshark [22] is used to capture them and converted into TCP and UDP flow conversations. The other set of data used is KDDCUP'99 [19, 23] dataset, which includes the malicious or intrusive network traffic data. KDD training dataset consists of approximately 4,900,000 records, each of which contains 41 features, labelled as normal or an attack. This data is in "pcap" format, which can be further analyzed by Wireshark. Wireshark is also used to analyze

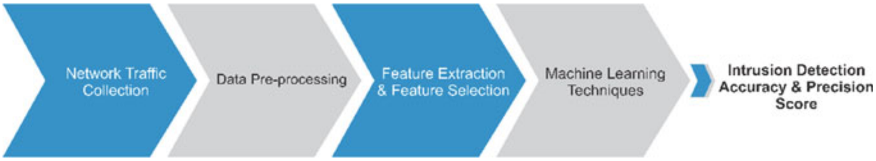
Network Traffic Collection
Data Pre-processing
Feature Extraction & Feature Selection
Machine Learning Techniques
Intrusion Detection Accuracy & Precision Score

Fig. 2 Proposed ıntrusion detection system model

network data, and then the data can be classified as normal and abnormal or intrusive data. Wireshark presents the data packets in a human-readable format, making it easier to understand the data better.

3.2 Data Pre-processing

The dataset is pre-processed, and in order to do that, the KDDCUP'99 data set is cleaned, and the redundant data is eliminated. The combination of both the datasets together is then divided into two sets of data, that is, training and testing datasets in the ratio of 70:30. These datasets, normal and intrusive datasets, before dividing them are labeled as "normal" and "attack" which helps distinguish the type of network traffic data.

3.3 Feature Extraction and Feature Selection

Based on [24, 25] and the survey conducted on the previous related work on the network traffic features pertaining to intrusion detection, techniques like PCA, LDA, etc. were used, and we found that 14 network traffic features extracted from the TCP and UDP flow conversations were of at most importance, represented in Table 1, in order to analyze the collected network traffic features for normal samples.

Out of these 14 traffic features, we had to obtain an optimal set of features to perform the machine learning techniques for detecting the intrusion. For this purpose, we used the feature selection technique, Feature Importance. Feature selection is a technique that reduces the amount of data to be analyzed. This is accomplished by identifying the most important features (or attributes) of a data set and discarding the less important ones. Feature importance renders a score for each of the features of the

Table 1 Network traffic features extracted

Network traffic features	
Average packet size	Average time between packets received
Average packet size received	Average time between packets sent
Average packet size sent	Ratio of ıncoming to outgoing bytes
Average flow duration	Ratio of ıncoming to outgoing packets
Number of packets	Bytes received
Rel start	Bytes sent
Duration	Number of bytes

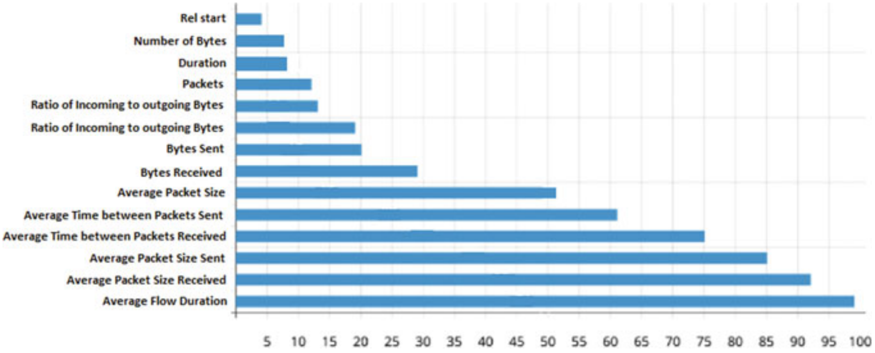

Fig. 3 Feature selection score graph

Table 2 Optimal network traffic features selected

Label	Network traffic features
F1	Average flow duration
F2	Average packet size received
F3	Average packet size sent
F4	Average time between packets received
F5	Average time between packets sent
F6	Average packet size

network traffic data; the higher the score more relevant or important is the feature. As we have implemented the intrusion detection model in Python, the importance selection is used, which is an in-built class.

Figure 3 depicts the Feature Selection score graph, which presents the least important to the most important network traffic features based on the importance score evaluated. Based on this feature selection score graph, we have set a threshold to 50; that is, we will select the most significant features that lie above the set threshold. So, all the features having a score greater than 50 are considered the most optimal feature set, represented in Table 2.

3.4 ML for Intrusion Detection System

After selecting the most optimal network traffic feature dataset, various machine learning techniques like Support Vector Machine (SVM), Logistic Regression, Naïve Bayes, Decision trees, K-Nearest Neighbour (KNN), and Random Forest are applied to these feature set in order to detect the intrusive traffic from the normal.

Support Vector Machine (SVM)

Support Vector Machine(SVM) - A Support Vector Machine(SVM) model is a supervised machine learning algorithm that analyzes the data used for the purpose of regression and classification. This algorithm aims to discover a hyper plain in N-dimensional space (where N is the number of features) that separately performs the data points classification. Support Vector Machine(SVM) [8] can perform linear and non-linear classification, implicitly mapping the inputs in the high-dimensional feature space. All flat affine subspaces are called hyperplanes. SVM Kernel is used to add more dimension to low dimensional space making it easier to segregate the data; it converts the inseparable problem to a separable problem by adding more dimension using kernel tree.

Decision Tree

A decision tree, as the name suggests, is a tree-like graph that has internal nodes that represent the test done on the attributes/features, and the branches show the decision rules of the test, leaf nodes which represent the outcome. Decision tree, which is a supervised machine learning algorithm, is used in making the classification and regression models.

Logistic Regression

Logistic Regression [26], being a supervised machine learning algorithm, is used in classification analysis, which helps in the prediction of variable data set probability. It assesses the interrelation between the dependent (Label) and the independent (Features) variable. Sigmoid function is used in the logistic function as a cost function. This logistic function helps map predictions to probabilities, and by fitting the data to this function, the probability of occurrence of an event can be predicted.

Naïve Bayes

The Naïve Bayes classifier is a probabilistic classifier that imposes a strong independence assumption [27], which suggests that the probability of an attribute doesn't affect the probability of the other. The dataset is converted into frequency tables, and further, a new table is generated on the basis of the evaluated probabilities of the features/attributes under consideration. For an n attributes series, the naïve Bayes classifier produces $2n!$ independent assumptions. Nevertheless, the Naïve Bayes classifier often provides correct results.

K-Nearest Neighbour (KNN)

K-Nearest Neighbour(KNN) is a supervised classifier, machine learning algorithm. KNN stores all the values present in the data set and classifies them into a new data point based on the character similarities. K-Nearest Neighbour (KNN) assumes that things with similar characteristics are near each other; that is, similar things exist in close proximity. The position where the target variable will be placed is predicted by finding the k closest neighbour, by calculating the Euclidean Distance.

Random Forest

Random Forest algorithm is mainly utilized for the purpose of classification analysis but can also be used for regression analysis. Different decision trees are created on various data samples, the prediction is taken from each of the decision trees, and then the voting is done to get the final prediction. Higher accuracy will be achieved by including a higher number of trees in the model.

4 Results and Discussions

In this section, we have implemented the Machine Learning (ML) algorithms stated in the previous section on the selected optimal set of network traffic features. A detailed analysis and comparison of these features on the basis of the ML algorithms have been drawn to depict their corresponding accuracy and precision.

4.1 ML Detection Accuracy on Individual Network Traffic Features

Table 3 presents the evaluated accuracy of ML techniques on individual network traffic features. İt is observed that when all the considered ML algorithms are applied to the individual features, it was observed that Random Forest has the highest detection accuracy, that is, an average accuracy of 97.85%.

Table 3 Detection accuracy (%) for ındividual features

Algorithm		$F1$	$F2$	$F3$	$F4$	$F5$	$F6$
Decision tree		88.72	89.23	89.74	91.73	81.02	87.72
Naïve Bayes		79.51	79.18	83.82	88.34	80.50	89.87
Random forest		88.89	99.10	99.85	99.96	99.68	97.45
SVM		85.29	95.88	94.43	90.22	99.98	97.45
Logistic regression		86.86	95.13	89.31	87.77	98.90	93.86
KNN	$K = 5$	84.85	99.35	99.30	99.51	99.89	98.85
	$K = 10$	85.37	99.34	99.30	99.50	99.87	98.59

4.2 ML Detection Accuracy on Combined Network Traffic Features

We now evaluate the detection accuracy for the combination of all the network traffic features, which is summarized in Table 4, representing the detection results for all six traffic feature combinations. İt is observed that when the ML algorithms are applied to the combination of features, Logistic Regression has the highest detection accuracy, that is, an average accuracy of 99.048%.

We observed that combining the optimal network traffic features leads to better intrusion detection accuracy. These results are concluded based on the traffic features we had selected. At the same time, if we include a traffic feature that has a network selection score less than (<) 50, we observed that the detection accuracy for the combined feature set of 7 features, that is including the feature *Bytes Received*, the intrusion detection accuracy is reduced in comparison to the detection accuracy of the combination of top 6 features, summarized in Table 5.

Table 4 Detection accuracy (%) for 6 combined features

Algorithm		$F1$ and $F2$ and $F3$ and $F4$ and $F5$ and $F6$
Decision tree		92.389
Naïve Bayes		88.487
Random forest		97.881
SVM		91.636
Logistic regression		99.048
KNN	$K = 5$	98.593
	$K = 10$	98.472

Table 5 Detection accuracy (%) for 7 combined features

Algorithm		$F1$ and $F2$ and $F3$ and $F4$ and $F5$ and $F6$ and $F7$
Decision tree		91.781
Naïve Bayes		88.394
Random forest		96.126
SVM		90.208
Logistic regression		99.012
KNN	$K = 5$	97.71
	$K = 10$	97.23

Table 6 Precision score on individual features

Algorithm		$F1$	$F2$	$F3$	$F4$	$F5$	$F6$
Decision tree		0.64	0.89	0.64	0.85	0.69	0.47
Naïve Bayes		0.89	0.73	0.64	0.92	0.66	0.43
Random forest		0.66	0.95	0.97	0.99	0.97	0.99
SVM		0.66	0.96	0.97	0.95	0.94	0.97
Logistic regression		0.91	0.87	0.98	0.96	0.93	0.99
KNN	$K = 5$	0.67	0.98	0.99	0.99	0.99	0.98
	$K = 10$	0.70	0.98	0.99	0.99	0.99	0.98

Table 7 Precision score on combined features

Algorithm		$F1$ and $F2$ and $F3$ and $F4$ and $F5$ and $F6$
Decision tree		0.86
Naïve Bayes		0.92
Random forest		0.97
SVM		0.96
Logistic regression		0.99
KNN	$K = 5$	0.98
	$K = 10$	0.98

4.3 ML Detection Precision on Individual Network Traffic Features

Comparison of Precision evaluated with the help of the stated ML algorithms on all the individual features are represented in Table 6.

Table 7 display the evaluated precision values when the ML algorithms were implemented on the combination of 6 features optimal feature set.

On performing the analysis, we observed that Logistic Regression had the highest average testing precision score of 0.94 when ML algorithms were implemented on individual features, and also, Logistic Regression outperformed the other ML algorithms when a combination of all the features was considered, with the highest precision score of 0.99. Once again, we can observe that we get a better precision score on combining the features than the individual features precision score.

5 Conclusion

In this paper, we compared the Machine Learning (ML) techniques on the normal and the intrusive network traffic dataset based on the detection accuracy and precision score. Once we extracted the network traffic features, we first selected the optimal set of features by performing the Feature Selection technique, Feature Importance. This project aims to find the optimal feature set, which would, in turn, provide us better detection results for the anomaly-based Network Intrusion Detection System with the help of the Machine Learning algorithms. On experimenting, we observed that the ML techniques performance was improved when implemented on the combination of network traffic feature set, that is, rather than implemented on the individual features. The highest accuracy of 99.048% and the highest precision score were achieved using the Logistic Regression technique when applied to the combined feature set. A detailed analysis of all the ML algorithms is also presented in our work. To the best of our knowledge, none of the existing work focuses on multiple supervised and unsupervised ML techniques. Also, the experiments in the existing work are performed on the standard dataset like KDD'99 or NSL-KDD datasets. We intended to select the most optimum feature set on the collected real-time normal dataset and perform the supervised and unsupervised ML techniques on them for effective intrusion detection. For our future work, we look forward to considering a larger, more extensive network traffic feature set in order to find a more optimal feature set for the improvement of intrusion detection. Perform intrusion detection based on the types of attacks involved in the network as our proposed work does not involve network attack classification, and also perform intrusion detection using a Deep Learning-based model.

References

1. Smys, S., Basar, A., Wang, H.: Hybrid intrusion detection system for internet of things (IoT). J. ISMAC **02**(04), 190–199 (2020)
2. Baraneetharan, E.: Role of machine learning algorithms intrusion detection in WSNs: a survey. J. Inf. Technol. Dig. World **02**(03), 161–173 (2020)
3. Sathesh, A.: Enhanced soft computing approaches for intrusion detection schemes in social media networks. J. Soft Comput. Paradigm (JSCP) **1**(02), 69–79 (2019)
4. Vengatesan, K., Kumar, A., Naik, R., Verma, D.K.: Anomaly based novel intrusion detection system for network traffic reduction. In: 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), pp. 688–690, Palladam, India (2018)
5. Gao, W., Morris, T.: On cyber attacks and signature based intrusion detection for modbus based industrial control systems. J. Dig. Forensics Secur. Law **9**(1), 37–56 (2014)
6. Jyothsna, V., Rama Prasad, V.V., Munivara Prasad, K.: A review of anomaly based intrusion detection systems. Int. J. Comput. Appl. **28**(7), 26–35 (2011)
7. Sinclair, C., Pierce, L., Matzner, S.: An application of machine learning to network intrusion detection. In: 15th Annual Computer Security Applications Conference (ACSAC'99), pp. 371–377, Phoenix (1999)
8. Mulay, S.A., Devale, P.R., Garje, G.V.: Intrusion detection system using support vector machine and decision tree. Int. J. Comput. Appl. **3**(3), 40–43 (2010)
9. Eesa, A.S., Orman, Z., Brifcani, A.M.A.: A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. Expert Syst. Appl. **42**(5), 2670–2679 (2015)
10. Kim, G., Lee, S., Kim, S.: A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Syst. Appl. **41**(4), 1690–1700 (2014)

11. Dreiseitl, S., Ohno-Machado, L.: Logistic regression and artificial neural network classification models: a methodology review. J. Biomed. Inform. **35**(5–6), 352–359 (2002)
12. Ghosh, P., Mitra, R.: Proposed GA-BFSS and logistic regression based intrusion detection system. In: 3rd International Conference on Computer, Communication, Control and Information Technology (C3IT), pp. 1–6, Hooghly (2015)
13. Bapat, R., Mandya, A., Liu, X., Abraham, B., Brown, D.E., Kang, H., Veeraraghavan, M.: Identifying malicious botnet traffic using logistic regression. In: Systems and Information Engineering Design Symposium (SIEDS), pp. 266–271, Charlottesville, VA (2018)
14. Bamakan, S.M.H., Wang, H., Tian, Y., Shi, Y.: An effective intrusion detection framework based on mclp/svm optimized by time-varying chaos particle swarm optimization. Neurocomputing **199**, 90–102 (2016)
15. Ahmad, I., Basheri, M., Iqbal, M.J., Rahim, A.: Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access **6**, 33789–33795 (2018)
16. Tao, P., Sun, Z., Sun, Z.: An improved intrusion detection algorithm based on GA and SVM. IEEE Access **6**, 13624–13631 (2018)
17. Farnaaz, N., Jabbar, M.: Random forest modeling for network intrusion detection system. Proc. Comput. Sci. **89**(1), 213–217 (2016)
18. Repalle, S.A., Kolluru, V.R.: Intrusion detection system using ai and machine learning algorithm. Int. Res. J. Eng. Technol. (IRJET) **4**(12), 1709–1715 (2017)
19. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: Knowledge discovery and data mining: towards a unifying framework. KDD **96**, 82–88 (1996)
20. Kim, D.S., Park, J.S.: Network-based intrusion detection with support vector machines. In: International Conference on Information Networking ICOIN 2003, Lecture Notes in Computer Science, pp. 747–756, Korea (2003)
21. Wang, H., Jie, Gu., Wang, S.: An effective intrusion detection framework based on SVM with feature augmentation. Knowl.-Based Syst. **136**, 130–139 (2017)
22. Gupta, S., Mamtora, R.: Intrusion detection system using wireshark. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **2**(11), 358–363 (2012)
23. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 data set. In: IEEE Symposium on Computational İntelligence for Security and Defense Applications, pp. 1–6, Otawa (2009)
24. Arora, A., Peddoju, S.K.: Minimizing network traffic features for Android mobile malware detection. In: 18th ACM International Conference on Distributed Computing and Networking ICDCN'17, no. 32, pp. 1–10, India (2017)
25. Arora, A., Peddoju, S.K.: Malware detection using network traffic analysis in android based mobile devices. In: 8th International conference on Next Generation Mobile Apps, Services and Technologies, pp. 66–71, India (2014)
26. Böhning, D.: Multinomial logistic regression algorithm. Annal. Inst. Stat. Math. **44**(1), 197–200 (1992)
27. Al-Sharafat, W.S., Naoum, R.: Development of genetic-based machine learning for network intrusion detection. Int. J. Comput. Inf. Eng. **3**(7), 1677–1681 (2009)