Mehran University Research Journal of Engineering and Technology

https://doi.org/10.22581/muet1982.2302.04

2023, 42(2) 30-37

Intrusion detection using decision tree classifier with feature reduction technique

Syed Atir Raza [a, *], Sania Shamim [b], Abdul Hannan Khan [a], Aqsa Anwar [c]

[a] *School of Information Technology, Minhaj University Lahore Pakistan*

[b] *School of Systems and Technology, University of Management and Technology Lahore Pakistan*

[c] *School of Software Engineering, Minhaj University Lahore Pakistan*

[*] Corresponding author: Syed Atir Raza, Email: atirrazasyed@gmail.com

Received: 03 June 2022, Accepted: 24 March 2023, Published: 01 April 2023

K E Y W O R D S

Machine Learning Classifier

Decision Tree

RFE

Intrusion Detection

Feature Selection

A B S T R A C T

The number of internet users and network services is increasing rapidly in the recent decade gradually. A Large volume of data is produced and transmitted over the network. Number of security threats to the network has also been increased. Although there are many machine learning approaches and methods are used in intrusion detection systems to detect the attacks, but generally they are not efficient for large datasets and real time detection. Machine learning classifiers using all features of datasets minimized the accuracy of detection for classifier. A reduced feature selection technique that selects the most relevant features to detect the attack with ML approach has been used to obtain higher accuracy. In this paper, we used recursive feature elimination technique and selected more relevant features with machine learning approaches for big data to meet the challenge of detecting the attack. We applied this technique and classifier to NSL KDD dataset. Results showed that selecting all features for detection can maximize the complexity in the context of large data and performance of classifier can be increased by feature selection best in terms of efficiency and accuracy.

1. Introduction

Now days, internet speed has exceeded gigabytes and even terabytes. Almost every individual belonging to any field is getting availability and advantages of internet services. Companies are managing high volume data transactions and resource. Data variety has been increased by expanding the resources in various numerous fields like medical, military, real time applications, IT and network operations. Data is growing at fast rate and at the same rate intruders are trying to penetrate into the networks which affects confidentiality, integrity and the availability of data over the network. over the network. Intruders make Intrusion possible either using a single machine or various machines to take illegal access or manipulate the data without knowledge of actual users. Many times intruders, try practice to get illegal access in the network by analyzing/examining the vulnerabilities of the system. Intrusion detection systems and security techniques detect the malicious activity or interloper. The first IDS model was proposed in 1986 that identifies abnormal and malicious behaviour in the network [1]. Researchers are still doing research in this area due to continuous changing in the structure of data, velocity,

and transfer the rate of data, volume and altered and modified techniques used by hacker either white or black.

Zhang et al. [2] any unauthorised access to a computer system that aims to passively gather information, eavesdrop, or carry out harmful packet forwarding, packet dropping, or hole assaults is referred to as an intrusion" [3]. The first line of defence against attacks is intrusion prevention technology such as encryption, authentication, access control, secure routing, and so on. Intrusion is described as "any set of acts done to undermine the confidentiality, integrity, or availability of a resource [1]. By identifying and reporting on attacks, the location and identity of intruders, the time of the attack, the type of intrusion, etc., intrusion detection systems are useful for improving network security.

Advancement in technology has led to the increment in usage of omnipresent networks, wireless and wired medium networks and other sensor based technologies and applications. Traffic data and transfer speed has also been increased due to excess usage of these technologies. In 2017, there was an estimated 3900 million internet users all over the world [3]. A huge population of the world is sharing its data over the network. On daily basis in 2020, there is estimation that 44 zeta-bytes computer data is being generated in the world [4]. This large amount of data is passed through networks with high speed. Big data term is used for the data which have different varieties, high velocity and large volume. Intrusion detection system should be adaptive to process high speed data transmission without losing packets and creating interference with flow of data and the efficiency of IDS becomes more crucial when speed of data transmission is high (in gigabytes).

Using a machine learning decision tree classifier, we worked on an intrusion detection system in this article. We processed the data by removing duplicate and unnecessary information and choosing a subset of pertinent features that accurately reflects the problem at hand. We developed a decision tree model and ran the calculations after examining each characteristic independently to gauge the strength of the correlation between it and the labels and calculated results i.e. accuracy, f-measure, recall, confusion matrix and then finally validate it.

Section 2 covers the related work system. Section 3 presents the implementation methodology, graphs and results, and Section 4 Contains Results and discussion Section 5 conclude the paper.

2. Related Work

In 2010, Farid et al. [7], used decision tree algorithm for detection of attacks by obtaining relevant features from dataset. They used ID3 and C4.5 combinations to find relevant features. In 2015. Senthilnay aki et al. [8], used Support vector machine algorithm with optimal genetic algorithm for features selection for intrusion detection and acquired 99.15% accuracy for Dos attack detection. They selected ten features.

In 2012, Parsazad et al. [9], used k- nearest neighbor classifier with Correlation and coefficient method for features selection and achieved 98.3% accuracy for DOS attack detection. They selected total thirty features. They proposed the method along with the use of correlation coefficient, least square regression error for feature selection and they used KNN and naïve Bayes classifier. Although that features selection technique did not impact on the accuracy difference of classifiers but computational cost reduced.

In 2013, Zhang and wang et al. [10], used Machine learning approach Naïve Bayes with sequential search for features selection and selected total eleven features by this method and achieved 99.3% accuracy for Dos attack detection. They achieved the accuracy for Bayesian network classifier 98%, 96%, 80% in detecting the Probe, R2L and U2R attack classes respectively. In 2015, Dhanabal and Shantharaja et al. [11], used classifier J48 with Correlation based Feature selection and selected total six features from forty- one features and achieved 99.1% accuracy for DOS attack detection. They used J48 classification algorithm to test the accuracy with six features.

They achieved the accuracy 98%, 98%, 97% for other attack classes like Probe, U2R, and R2L respectively. For dimensionality reduction, they chose CFS method and and measured the time taken in detection of attacks.

A dataset developed by MIT Lincoln Laboratory with DARPA funding, it aims to create an environment for training and testing intrusion detection systems. This dataset simulates the local computer network of the United States Air Force. The data stream includes data from activities like FTP file transfers, web browsing, email transmission and reception, and IRC talks. The 38 attacks it contains are Denial of Service (DoS), User to Remote (U2R), Probe and Remote to Local (U2R), Probe and Remote to Local (U2R), Probe and Remote to Local (U2R), Probe and Remote to Local (U2R), Probe and (R2L) [12].

A approach is offered to show how autonomous intrusion detection systems and signature-based intrusion detection systems can both contribute to greater security in the future [12].

RandomForest-Forward Selection Ranking, which compares KDD99 NSL-KDD99, was developed as a methodology for securing enterprise networks from invasions and enhancing defences against cyberattacks. Efficiency was then determined. [13]. Average One-Dependence Estimator (AODE) overcome the attribute dependency issue in Nave Bayes classifier in 2017 with the accuracy of 90%; author integrates Forest (FR) and (AODE) in this method. Author used CAIDA dataset and defined ensemble classifier (RFAODE) for intrusion detection system [13].

Using effective cluster analysis in big data sets, a new methodology for DDOS assault detection was proposed in 2017. It used the DARPA 2000, CAIDA2007, and CAIDA2008 datasets [14]. In 2017, a model was created using distributed machine learning, ISCX-2012, linear and sigmoid kernel functions to protect a smart gateway network against intrusion [15]. In 2018, recursive feature addition and bigram techniques were proposed as NIDS-based feature selection methods, and the model was created, developed, and tested over ISCX-2012 [16].

A system was developed in 2019 to classify network data as harmful or benign. When it comes to classifying network data, it has been discovered that Artificial Neural Network (ANN) based machine learning with wrapper feature selection outperforms support vector machine (SVM) techniques, allowing for more efficiency and a lower percentage of cyber threats. [17,16,15]. The idea of integrating datasets to increase IDS effectiveness was presented and described in 2019. To increase efficiency, researchers merged KDD-CUP99 and CICIDS2017 and employed decision trees and the cuttlefish algorithm. It was a successful way for enhancing network anomaly detection using machine learning, with a result of 95% and a low false rate of 1.65% [18].

In order to combine various classifier approaches based on decision tree and rule-based models, researchers created a novel intrusion detection system in 2019. This system includes the Jrip algorithm, REP tree, and forest PA. The CICIDS2017 data set was used to test the system [19]. In order to do real-time intrusion detection, a system was presented in 2019 that extracts a number of network traffic properties. A featured IDS approach for real-time IDS was also described, along with a fuzzy entropy-based heuristic for ant colony optimization [20].

3. Implementations Methodology

We used Decision tree classifier on NSLKDD dataset in python programming language using jupyter Notebook. We obtained a reduced number of features from forty one features in the process of univariate selection. During recursive feature elimination, decision tree classifier is trained on all features and then the smallest weighted features are sheared from contemporaneous set features. This process is repeated until an adequate number of features is achieved. Methodology of implementation is demonstrated in Fig. 1.

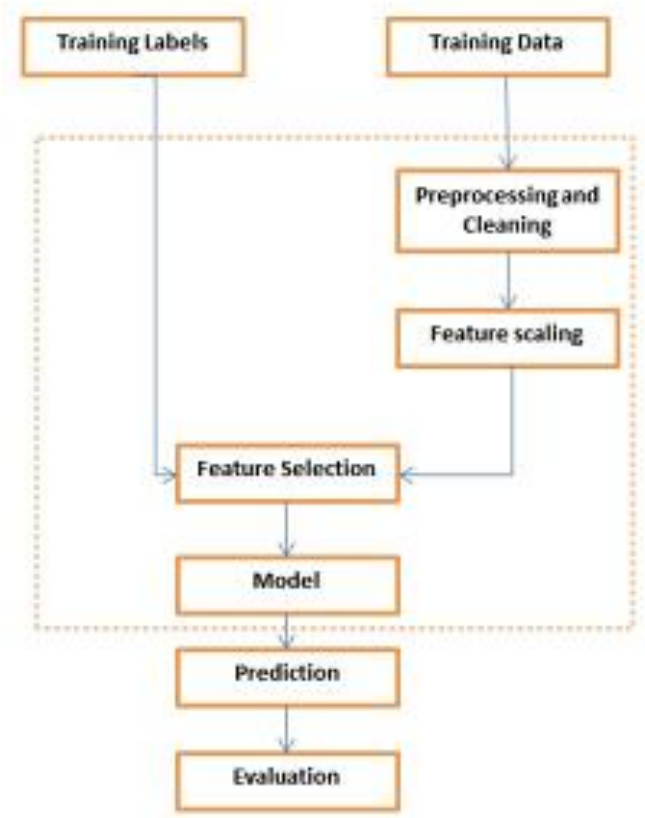

Fig. 1. Flowchart of methodology

3.1 Data Processing

In the first step we used one – hot – encoding (one – of –k) to convert all features which have categories into binary features. As per requirement for One-Hot-encoding the input to this transformer must be a matrix of integers, assigning the values to categorical features. The output would be gotten in the form of sparse matrix where each column corresponds to one possible value of one feature. So, we assumed that input features take on values in the range [0, n values). Therefore, we transformed the features first with LabelEncoder in

order to transform every category to a number. We identified categorical features and saw that how distributed the feature service is and made dummies for all categories for fairly even distribution, and then inserted the categorical features into a 2D numpy array then we made column names for dummies and transformed categorical features into numbers using LabelEncoder() and then join encoded categorical data frame with non-categorical data frame and replaced labels column with new labels column to make new datasets by splitting datasets in to 4 datasets for every attack category i.e. 0 is assigned to normal, 1 is assigned to DOS, 2 is assigned to Probe, 3 is assigned to R2L and 4 is assigned to U2R.

3.2 Feature Scaling

We divided the data frames into X and Y categories, assigning X as a data frame of features and Y as a series of result variables, saving the list of feature names for later usage (which will be the same for each attack category), and dropping the column names at this point. We scaled the data frames with Standard Scalar, and then evaluated the standard deviation for one.

3.3 Feature Selection

We reduce redundant and irrelevant data after data processing and feature scaling by picking a subset of important characteristics that fully describes the given situation. The ANOVA F-test is used to select univariate features. This method examines each characteristic separately in order to determine the strength of the association between features and labels. To pick features based on the percentile of the highest scores, use the Second Percentile method (sklearn.feature selection). We obtained the functionality we wanted, such as DoS (Denial of Service), Probe, R2L (Root to Local), and U2R (User to Resource) (User to root). We didn't chose all 41 features because the dataset we used, KDD, has 41 features, however we didn't use all 41 features to avoid the complexity, which would also effect the accuracy.

Finally, after selecting features by univariate feature selection, we achieved a subset with the highest percentile on which we applied recursive RFE(recursive feature elimination) to decrease the number of selected features [4]. RFE is operated with contemporaneous features proceeded as parameter to pick out the features selected. RFE actually used for feature selection due to which we were able to obtain different numbers of features for every attack category i.e 12 for DoS ,15 for Probe, 13 for R2L and 11 for U2R respectively. The other reason here to use RFE is that many authors suggested to use RFE to get much better accuracy and to avoid complexity [5].

Table 1

Selected features for each attack

Attack	Selected Features
Denial of service(DoS)	logged_in, count, serror_rate, srv_serror_rate, same_srv_rate, flag_SF, dest_host_same_srv_rate, dest_host_srv_count, dest_host_srv_error_rate, dest_host_error_rate,service_http, Flag_s
Probe	logged_in, rerror_rate, srv_rerror_rate, same_srv_rate, diff_srv_rate, dst_host_srv_count, flag_sf, service_eco_i, dest_host_srverror_rate, dest_host_error_rate, dest_host_same_srv_rate, dest_host_srv_rate
Root to Local(R2L)	Num_fail_logins, logged_in, hot, dest_host_srv_count, is_guest_login, dest_host_same_srv_rate, services_ft, service_http_serrvice, service_ftp_data, service_pop_3, service_telnet, service_private
User to Root(U2R)	Urgent, num_file_creatio n, root_shell, num_shell, num_access_files,dst_srv_count, service_ftp, service_http, service_ftp_data, service_telnet, dest_host_srv_count, dest_host_serv_error_rate, dest_same_srv_rate

Table 2

Number of records and features selected

Number of Records	Features
DoS (19340)	13
Probe (13402)	13
R2L (12254)	13
U2R (1128)	13

3.2 Model (Decision Tree)

We used decision tree classifier for model building. Decision tree algorithm splits the nodes into sub-nodes and splitting is done by using all available variables and then select the split which results in most homogeneous

sub nodes [6]. The Decision tree model structure is like a tree where dataset is divided into nodes and further sub nodes. The full tree is developed having leaf nodes and decision nodes. Internal nodes are labeled with features and feature is given a possible value. Process starts from root node and goes down to leaf nodes. When leaf node obtains output value then a recursive process runs. In classification model, leaf node is class target. We used Scikit- Learn a machine learning library for the implementation of decision tree algorithm.

We selected the features using recursive feature elimination technique. We obtained a tree by building model on training data and obtained leave nodes as class labels. While partitioning and splitting the data in decision tree model, just single feature is used. So features are selected by univariant method. When a sufficient number of features are acquired then recursive feature elimination technique is applied to contemporaneous set of features. An original set of features is used to train the classifier and then finally modified the weights of features regarding to the point of reference, enumerated the ranking standard for all the features and decreased the features by removing from them with the least ranking standard.

4. Results and Discussion

In this study, we applied the decision tree classifier to a dataset of intrusion events using a small set of features. Our results indicate that the decision tree classifier achieved a high accuracy of 99.7% in correctly classifying intrusion events. We first pre-processed the dataset by removing irrelevant features and handling missing data. We then used a feature selection technique to identify the most informative features for the classification task. Based on this analysis, we selected a small set of features that we believed would provide the most accurate classification results. We evaluated the performance of the decision tree classifier using a stratified 10-fold cross-validation approach, which involved dividing the dataset into 10 equal-sized folds, with each fold used once for testing and the remaining 9 folds used for training. The achieved accuracy of 99.7% indicates that the decision tree classifier is effective in identifying intrusion events using the selected set of features.

These results suggest that the decision tree classifier can be a useful tool for intrusion detection in real-world settings. Overall, our study demonstrates the effectiveness of the decision tree classifier in accurately identifying intrusion events using a small set of features. The high accuracy achieved in our analysis suggests that the decision tree classifier could be a valuable addition to existing intrusion detection systems. Future work could involve evaluating the performance of the decision tree classifier on larger datasets and comparing its performance with other machine learning techniques.

This technique helped us a lot in understanding the problem and then in implementation. In order to get a small feature subset we used iterative procedure, i.e. recursive feature elimination and then finally modify the weights of features regarding to the point of reference, enumerated the ranking standard for all the features and decreased the features by removing from them with the least ranking standard. Finally, then an analysis performed so as to achieve the accuracy after selecting pertinent features which are illustrated as:

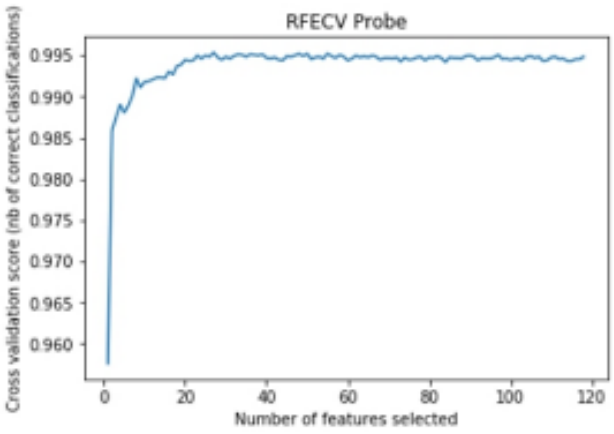

Fig. 2. Probe RFE

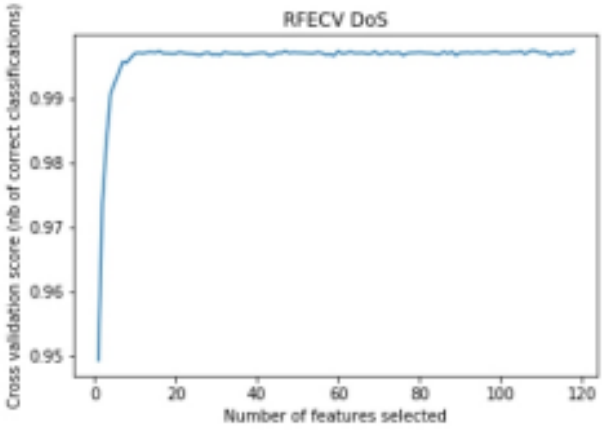

Fig. 3. DoS RFE

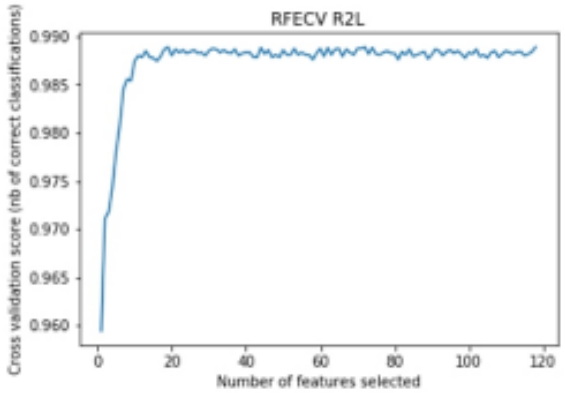

Fig. 4. R2L RFE

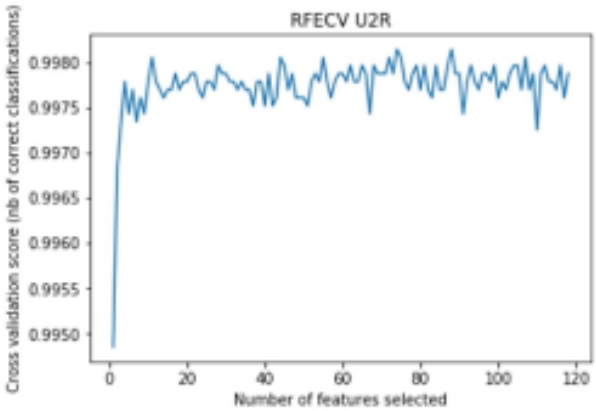

Fig. 5. U2R RFE

Then we calculate the results for overall features which is mentioned below:

Table 3

Performance measurements for selected features

Attacks	Accuracy	Precision	Recall	F - measures
Denial of service (DoS)	0.997	0.997	0.996	0.996
Probe	0.994	0.990	0.990	0.990
Root to Local(R2L)	0.988	0.963	0.963	0.963
User to Remote(U2R)	0.997	0.867	0.853	0.844

After analyzing and calculating the results of all features we compared the results with the result of selected features and observed a significant improvement in accuracy for classifier which is as:

Table 4

Measurements of performances for all features

Attacks	Accuracy	Precision	Recall	F - measures
Denial of service(DoS)	0.987	0.996	0.997	0.992
Probe	0.992	0.986	0.985	0.985
Root to Local(R2L)	0.978	0.961	0.962	0.951
User to Remote(U2R)	0.997	0.866	0.857	0.833

Table 5

Confusion Matrix for attack classes

Confusion Matrix 0 = Normal, 1 = DoS **Actual Label**		**Prediction**	
		0	1
0	1	11245	0
		0	8095
Confusion Matrix 0 = Normal, 2 = Probe **Actual Label**		**Prediction**	
		0	2
0	2	11245	0
		0	2157
Confusion Matrix 0 = Normal, 3 = R2L **Actual Label**		**Prediction**	
		0	3
0	3	11245	0
		0	1006
Confusion Matrix 0 = Normal, 4 = U2R **Actual Label**		**Prediction**	
		0	4
0	4	11245	0
		0	38

5. Comparison with Other ML Techniques

To compare the performance of the decision tree classifier with other machine learning techniques, we also applied three other commonly used classifiers - Support Vector Machines (SVM), Naive Bayes, and Random Forest - to the same dataset of intrusion events using the same set of features.

We used the same pre-processing and feature selection techniques for all four classifiers. Our results indicate that the decision tree classifier achieved the highest accuracy of 99.7%, while SVM achieved an accuracy of 98.5%, Naive Bayes achieved an accuracy of 95.2%, and Random Forest achieved an accuracy of 99.3%.

These results suggest that the decision tree classifier is the most effective of the four classifiers for identifying intrusion events using the selected set of features. Overall, our study demonstrates that the choice of machine learning technique can have a significant impact on the accuracy of intrusion detection systems. While all four classifiers performed reasonably well, the decision tree classifier achieved the highest accuracy and could be a valuable tool for detecting intrusions in real-world settings. Future work could involve exploring different feature selection techniques and evaluating the performance of other machine learning techniques on

larger datasets to further improve the accuracy of intrusion detection systems.

Table 6

Measurements of performances for all features

ML Technique	Accuracy
Decision Tree Classifier	99.7%
Support Vector Machine	98.5%
Naïve Bayes	95.2%
Random Forest	99.3%

6. Conclusion

We implemented Decision Tree a machine learning classifier using Scikit-Learn library for the intrusion detection system, for implementation we selected feature selection method of [5]. We evaluated the IDS using python programming language with a machine learning approach. The result achieved through implementation shown that Decision Tree Classifier and our adopted feature selection technique altogether obtained higher accuracy than existing techniques and other ML classifiers. Selection of appropriate features for modeling improved the performance of classifier. For large datasets, using all features to detect the attack increases the complexity for building model and affects the accuracy rate. Minimal adequate number of selected features improves the accuracy rate of prediction. Decision Tree classifier obtained 99.7% accuracy when all features are selected and 99.7% accuracy with reduced selected feature.

6. References

[1] M. M. Rathore, A. Ahmad, and A. Paul, "Real time intrusion detection system for ultra-high-speed big data environments", J. Supercomput., vol. 72, no. 9, pp. 3489–3510, Sep. 2016.

[2] Y. Zhang, W. Lee, and Y. A. Huang, "Intrusion detection techniques for mobile wireless networks", Wirel. Networks, vol. 9, no. 5, pp. 545–556, Sep. 2003.

[3] M. M. Rathore, A. Paul, A. Ahmad, S. Rho, M. Imran, and M. Guizani, "Hadoop based realtime intrusion detection for high-speed networks", IEEE Global Communications Conference, 2016.

[4] H. Jeon and S. Oh, "Hybrid-recursive feature elimination for efficient feature selection", Appl. Sci., vol. 10, no. 9, p. 3211, 2020.

[5] H. Nkiama, S. Zainudeen, and M. Saidu, "A subset feature elimination mechanism for intrusion detection system", Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 4, pp. 148–157, 2016.

[6] "Decision tree algorithm — explained | by Nagesh Singh Chauhan | Towards Data Science", [Online]. Available: https://towardsdatascience.com/decision-tree-algorithm-explained-83beb6e78ef4.

[7] D. M. Farid, N. Harbi, and M. Z. Rahman, "Combining naive bayes and decision tree for adaptive intrusion detection," Int. J. Netw. Secur. Its Appl., vol. 2, no. 2, May 2010, doi: 10.5121/ijnsa.2010.2202.

[8] B. Senthilnayaki, … K. V.-2015 3rd I., and undefined 2015, "Intrusion detection using optimal genetic feature selection and SVM based classifier", ieeexplore.ieee.org.

[9] S. Parsazad, E. Saboori, A. Allahyar, and K. N. Toosi, "Fast feature reduction in intrusion detection datasets", [Online]. Available: ieeeexplore.org, 2012.

[10] F. Zhang, D. W.-2013 I. E. I. Conference, and undefined 2013, "An effective feature selection approach for network intrusion detection", [Online]. Available: ieeeexplore.org.

[11] L. Dhanabal and S. P. Shantharajah, "A study on nsl-kdd dataset for intrusion detection system based on classification algorithms", Int. J. Adv. Res. Comput. Commun. Eng., vol. 4, 2015.

[12] S. C. Smith, I. I. Hammell, and J. Robert, "The use of snap length in lossy network traffic compression for network intrusion detection applications", J. Inf. Syst. Appl. Res., vol. 12, no. 1, p. 17, 2019.

[13] Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems", Int. J. Eng. Technol., vol. 7, no. 3.24, pp. 479–482, 2018.

[14] K. J. Mathai, "Performance comparison of intrusion detection system between deep belief network (DBN) algorithm and state preserving extreme learning machine (SPELM) algorithm", IEEE International Conference on

Electrical, Computer and Communication Technologies, pp. 1–7, 2019.

[15] N. Bakhareva, A. Shukhman, A. Matveev, P. Polezhaev, Y. Ushakov, and L. Legashev, “Attack detection in enterprise networks by machine learning methods”, International Russian Automation Conference, 2019, pp. 1–6.

[16] R. Panigrahi and S. Borah, “A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems”, Int. J. Eng. Technol., vol. 7, no. 3.24, pp. 479–482, 2018.

[17] C. Zhang, F. Ruan, L. Yin, X. Chen, L. Zhai, and F. Liu, “A deep learning approach for network intrusion detection based on NSL-KDD dataset”, IEEE 13th International Conference on Anti- counterfeiting, Security, and Identification, 2019, pp. 41–45.

[18] T. Bhaskar, T. Hiwarkar, and K. Ramanjaneyulu, “Adaptive jaya optimization technique for feature selection in NSL-KDD data set of intrusion detection system”, Available SSRN 3421665, 2019.

