International Journal of Safety and Security Engineering
Vol. 14, No. 5, October, 2024, pp. 1431-1437
Journal homepage: http://iieta.org/journals/ijsse

Performance Exploration of Network Intrusion Detection System with Neural Network Classifier on The KDD Dataset

Sellappan Devaraju[1], Dheresh Soni[1], Sundaram Jawahar[2], Jay Prakash Maurya[1], Vipin Tiwari[3*]

[1] School of Computing Science and Engineering, VIT Bhopal University, Bhopal 466114, India
[2] Department of Computer Science and Applications, Christ Academy Institute for Advanced Studies, Bangalore 560083, India
[3] Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, India

Corresponding Author Email: vipintiwari1@gmail.com

Copyright: ©2024 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/ijsse.140510

Received: 20 December 2023
Revised: 26 August 2024
Accepted: 9 September 2024
Available online: 31 October 2024

Keywords:
MATLAB, network Intrusion detection, KDD dataset, neural networks

ABSTRACT

Network Intrusion Detection Systems (NIDS) are a difficult task for determining in any managerial information system or IT sectors, if a user is a normal user or an attacker. The main objectives of the proposed system are to enhance operational efficiency, decreasing the occurrence of false positives, to minimize the time complexity of the process. It is an excellent way for dealing with various types of network problems. Research focusses the various classifiers are applied to detect various types of network assaults. Performance of network intrusion detection by two classifiers are used to compare the results. Probabilistic Neural Network (PNN) and Feed Forward Neural Network (FFNN) classifiers are employed this suggested study. The performance results comparison between full featured and reduced features are presented. MATLAB software application is applied to test the performance of both test and train dataset. Detecting network intrusions is a critical challenge within managerial information systems and the IT sector, as it involves the complex task of distinguishing between legitimate users and potential attackers. Maintaining a secure network environment is paramount to safeguarding sensitive information and operations. In the arena of network intrusion detection, the research predominantly revolves around the deployment of diverse classifiers to identify various types of network attacks. This paper, proposes the evaluation of two specific classifiers, the PNN and the FFNN, with the objective of comparing their performance in the context of network intrusion detection. We systematically assess their effectiveness in both full-featured and reduced-feature scenarios, utilizing MATLAB software to rigorously analyze their capabilities across test and training datasets. In essence, this research delves into the intricate realm of Network Intrusion Detection Systems (NIDS), investigating how the PNN and FFNN classifiers function in the critical role of safeguarding networks against a multitude of potential threats. Through comprehensive analysis, we aim to illuminate the most efficient approach to enhancing network security in the constantly evolving landscape of cybersecurity. As a result, it is recommended that FFNN approaches be adopted as a means of improving detection efficiency and reducing the False Positive Rate (FPR) in network intrusion detection systems.

1. INTRODUCTION

One security management technique is the Network Intrusion Detection System (NIDS). Anti-thread software is placed on the server in NIDS to check if any threads are entering and exiting the system. Information protection is crucial for all organizations, agencies of the government, educational institutions, and other relevant industries involved in daily operations. NIDS is divided into two categories: (i) signature-based, which compares an intrusion to the log file already present in the database to detect it; and (ii) anomaly-based, which monitors system activity and classifies it as either attack or normal using heuristics rather than signatures or patterns to detect misuse and computer intrusions. NIDS is further divided into two categories: (i) host-based, which tracks only the device's incoming and outgoing packets; and (ii) network-based. The signature-based intrusion detection system compares a new signature with one that already exists in the database to identify intrusions. If the attack that was found and its matching signatures match those in the database, it's considered an incursion. Unidentified assaults are intrusion detection attempts that are detected on a network because they deviate from standard attacks and are based on anomalies. Systems for detecting network intrusions separate host-based threats from network-based attacks. Interconnected computer systems are able to identify network-based threats. Only one computer can be used to identify and stop host-based assaults. assaults using web-based techniques are feasible when systems are connected to the internet. These assaults can spread to other systems through email, chat, information

downloads, and other means. Hazardous web-based attacks can now target a large number of computer systems [1].

Through the Internet, users actively contribute important knowledge, offering many chances to improve the security and stability of networked systems. NIDS are essential for classifying successful intrusions and enabling prompt countermeasure deployment to stop or track attacks. There are two main kinds of network attacks. By connecting network activity to predetermined signatures kept in a database, signature-based intrusion detection systems identify intrusions. Activity that is found and that matches these signatures is considered perhaps intrusive. Anomaly-based intrusion detection systems, on the other hand, distinguish unexpected threats by departing from standard network activity patterns. NIDS are made expressly to recognize and separate host-based threats from network-based attacks. Computer systems are vulnerable to network-based attacks because of their interconnectedness. These attacks can enter systems through various web-based channels, including email, chat services, downloads, and more. As a result, a lot of computer systems are now susceptible to potentially dangerous online attacks [2].

In conclusion, the abundance of knowledge exchange on the Internet has produced a setting where network security is critical. By classifying attacks into misuse and anomaly-based categories and protecting networked systems from the constantly changing web threat landscape, NIDS play a crucial role as protectors, guaranteeing the stability and security of these systems.

2. LITERATURE REVIEW

Neural Network is the effective way to detecting the intrusion more effectively. To address these issues, we propose the utilization of FFNN and PNN classifiers for the purpose of detecting intrusions based on signatures. Multiple tactics are utilised in the MATLAB programme in this challenge to increase performance on the KDD dataset. Performances are compared with full-featured and reduced features as well. FFNN has been classified into two distinct categories, namely single-layer and multi-layer. The single-layer type is the initial and fundamental form of independent machine learning, comprising solely of input and output layers. On the other hand, the multi-layer type consists of three layers, namely input, hidden, and output layers. The Forward Phase is utilized to establish fixed parameters and culminates with the error signal. During Backward Phase, incorrect signals are transmitted via network. During this phase, the network's free parameters are adjusted to decrease statistical error. The PNN is a natural extension of the Bayes classifier work. To be more exact, it is considered the function that approximates distribution's probability density. PNN are grouped into pattern, summation and output layer. Proposed technique utilised to classify KDD Dataset. The four classifications that are classified and compared are Normal with DoS, Probe, R2L, and U2R. The MATLAB code is used to calculate precision, recall, F-value, and efficiency. Network intrusion detection system is crucial in identifying real-world intrusions. As a result of the numerous incursions that have plagued many systems, a variety of strategies and tactics have been offered. The identification of intrusions is facilitated through the utilization of various methodologies such as data mining, neural networks, and statistical algorithms. The present paper is interrelated with the discussion of diverse approaches and techniques that are employed for the purpose of detecting intrusions. The topics of efficiency and accuracy are addressed by the Conditional Random Fields (CRF) and Layered Approach. This methodology employs the utilization of CRF and Layered techniques to effectively and precisely identify attacks. The approach in question is capable of detecting attacks through the use of the KDD dataset [1].

To reduce the number of required resources to identify as an attack, Neural Network and algorithm of Rough Set were utilized [2, 3]. The data is tested using the KDDCup'99 dataset, producing a more robust and reliable conclusion. The utilization of data mining methodologies, such as decision trees, has proven to be effective in the identification of assaults [4]. The KDD dataset is employed for both training and testing purposes. This approach has demonstrated advancements in the detection of novel anomalies [5]. The identification of anomalies is ascertained through the utilization of Multivariate Statistical Analysis methodologies. Statistical techniques are employed to assess the aforementioned [6, 7]. Hidden Markov Model applied to provide anomaly intrusion detection via system calls [8]. To detect network intrusions, the Genetic Algorithm is used. The Genetic Algorithm considers information about spatial and temporal when encoding the problem. Genetic Algorithm is useful to identifying anomalies [9]. In order to decrease the computational intensity, various techniques for feature reduction are employed to diminish the amount of time required for calculations, while simultaneously enhancing accuracy [10]. The application of statistical classification techniques, specifically the identification of anomalies through network-based analysis using the Hierarchical Gaussian Mixture Model, is being explored. The widely recognized KDD dataset has been selected as the basis for evaluating this model. To ascertain its feasibility and effectiveness, six categorization strategies have been employed for validation purposes [11]. The aforementioned methodology is employed in Intrusion Detection Systems with the aim of mitigating the occurrence of erroneous alerts and enhancing the precision of attack identification [12, 13].

The Correlation Coefficient Matrix is used to characterise normal or abnormal network traffic to accurately categorize several anomaly behaviours (DDoS attacks and network scanning) [14, 15]. The various techniques are analyzed with different features [16, 17]. The proposed system has been designed to address the limitations of the existing system and aims to mitigate the issues associated with the KDD dataset through the utilization of a neural network classifier [18, 19]. The proposed system aims to achieve the following objectives:

i. In order to enhance operational efficiency,
ii. With the aim of decreasing the occurrence of false positives,
iii. To minimize the time complexity of the process.

The subsequent content of this document comprises of the following: Section 2 delves into the discourse of pertinent literature. Section 3 furnishes details pertaining to the KDD dataset. Section 4 proffers experimental data, along with a corresponding discussion. Finally, Section 5 culminates with a conclusive summary.

3. KDD DATASET DESCRIPTION

In order to test intrusion detection systems, the KDD dataset

was employed. The training dataset comprises approximately 4,900,000 records with 41 attributes, either normal or attack [2]. KDD dataset contains various attacks and 24 different training assault. Four categories of attacks are represented [1, 8]. In KDD Dataset is categorized into many attacks, these attacks are divided into different classes. As per the nature of the attacks, it can be grouped into various classes. The KDD Dataset itself, these groups are categorized.

Denial of Service (DoS) attacks: For instance, they can prevent a system from responding to valid requests. Deluge [15, 20].

User-to-Root (U2R) attacks: These refer to instances of unapproved access to the root (local super user) rights. Overflow of buffer attacks.

Remote-to-Local (R2L) attacks: R2L assaults, or illegal access from a remote machine, are one type of attack which makes an incorrect password guess.

Probing: As an example, consider surveillance and other probing assaults. scanning ports.

In that sequence, sets are labelled AA, BB, CC, DD, and EE. The data for the set 'AA' is obtained from the DoS class. Set 'BB' gathers information from U2R. Set 'CC' is obtained from R2L. Set 'DD' gathers information from Probe. Set E'E' comes from Normal. The next data sets can be used to train and evaluate the KDD dataset data [21].

To detect assaults in the KDD dataset, 41 features and reduced features are applied in each class. The webpage lists the 41 features.

Every symbol is allocated an integer code to facilitate its conversion into numerical representation. For example, in the protocol_type, '0' is allocated to tcp, udp is 1, and icmp is 2. Initially, the attacks were categorized as follows [22]: 'AA' for Denial of Service (DoS), 'BB' for User to Root (U2R), 'CC' for Remote to Local (R2L), 'DD' for Probe, and 'EE' for Normal. The features src_bytes and dst_bytes possess an extensive integer range spanning from 0 to 1.3 billion. To limit this range to [0.000, 9.140], these attributes underwent logarithmic scaling with a base of 10. All other features are Boolean [23], with values ranging from [0.0, 1.0]. In result, scaling was unnecessary for these properties. Table 1 shows the category-wise attacks.

Table 1. Displays attacks classified by category

DoS	R2L	U2R	Probe
	guess_passwd		
smurf	ftp_write		
back	imap	loadmodule	nmap
pod	phf	buffer_overflow	ipsweep
land	multihop	rootkit	satan
teardrop	spy	perl	portsweep
neptune	warezmaster		
	warezclient		

Table 2. Data set for training and testing

	Training Set (10%)	Testing Set (Corrected)
DoS	391458	229853
U2R	52	70
R2L	1126	16347
Probe	4107	4166
Normal	97278	60593
Total	**494021**	**311029**

For training the network, the 10% KDD data is trained using two different neural networks. Generally, the networks are programmed to execute functions such as identifying patterns and making decisions. The training set is shown in Table 2. For network testing, the corrected KDD data is tested using two distinct neural networks. The accuracy of each neural network is verified by examining the KDD data. The testing set is shown in Table 2.

4. METHODOLOGY

4.1 Normalization

In order to normalize feature values, each feature's values are statistically analyzed using the KDD Cup 1999 dataset's available data. An acceptable maximum value is then established for each feature. The normalization of feature values in the interval [0, 1] is computed based on the maximum values and the subsequent straightforward formula.

$$\begin{aligned} &If\,(f\_V > MaximumF)\\ &N\_f = 1;\\ &Otherwise\\ &\mathrm{N}f = (\mathrm{f\ V} / MaximumF) \end{aligned} \tag{1}$$

F: Feature f_v: Feature value
MaximumF: Maximum acceptable value for F
N_f: Normalized or scaled value of F

4.2 Probabilistic neural network (PNN)

Directly following the work on Bayes classifiers is the PNN. To be more exact, the PNN is understood as a function that roughly represents the distribution's probability density. The PNN is made up of nodes that are arranged in the pattern, summation, and output layers-the three levels that come after the input layers.

Pattern Layer: One pattern node is used for every training stage. Every pattern node is a product of the weight vector and a specific node whose weights enter it for classification. The product is then run via the activation function after that:

$$\exp[(x^{\mathrm{T}}W_{ki} - 1)/\sigma^2] \tag{2}$$

Summation Layer: The following outputs are sent to each summation node by pattern nodes connected to a certain class:

$$\sum_{i=1^{Nk}} \exp\left[\frac{(x^T w_{ki} - 1)}{\sigma 2}\right] \tag{3}$$

Output Layer: Binary neurons serve as the output nodes, producing the categorization decision.

$$\sum_{i=1^{Nk}} \exp\left[\frac{(x^T w_{ki} - 1)}{\sigma 2}\right] > \sum_{i=1^{Nj}} \exp\left[\frac{(x^T w_{kj} - 1)}{\sigma 2}\right] \tag{4}$$

The smoothing factor, or the deviation of the Gaussian functions, is the only element that needs to be chosen for training. Excessive tiny deviations result in an extremely spiky approximation that is poorly suited for generalization, whereas

excessively large deviations smooth out details. An experiment determines which deviation is appropriate.

4.3 Feed Forward Neural Network (FFNN)

Only input to output signal transmission is possible with the FFNN. Generally speaking, FFNNs are simple networks that link inputs and outputs. They are widely applied in the field of pattern recognition. The FFNN are divided into two categories: Single-layer and multi-layer.

The first and most basic learning machine is the single-layer neural network. The input layer and output layer are examples of the two layers that make up the single layer. Three levels make up multi-layer feed forward networks: The input layer, the hidden layer, and the output layer.

In multilayer FFNN, there are two different phases that are used. The Forward Phase fixes the network's free parameter and computes the error signal at the end.

$$e = di - yi \tag{5}$$

where, yi is the network's actual output in response to the input, and di is the anticipated response. The network propagates the error signal (e.i.) during the Backward Phase. In order to statistically minimize the error e.i., the network's free parameters are adjusted at this phase.

5. EXPERIMENTS AND RESULTS

The layered technique's initial goal was to reduce calculation time in order to find abnormal events. Every layer of the model undergoes independent training before being deployed successively. The four layers are the probe, DoS, R2L, and U2R layers. Then, each layer is trained independently using a small set of essential features.

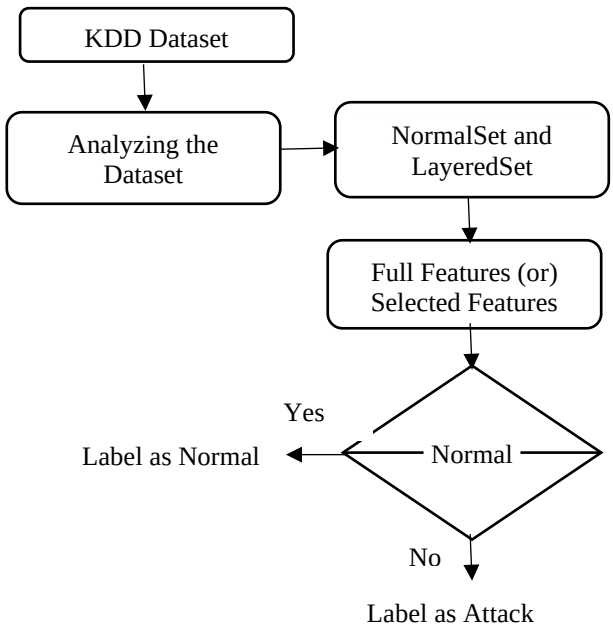

Figure 1. Depiction of an approach

Figure 1 illustrates a depiction of an Approach, wherein layer1 pertains to DoS, layer2 pertains to Probe, layer3 pertains to R2L, and layer4 pertains to U2R.

A layered classifier is suggested to identify the different types of classes. There are types of classes-DoS, Probe, R2L, U2R and Normal-are taken into consideration. There are various assaults in each class. The layered strategy is suggested to determine the class type; if the class is determined, attacks are detected; if not, the subsequent class is taken into account, and the procedure is repeated. To increase the detection rate and decrease computation time, layered classifiers are used specific attributes that are used to detect different types of attacks. Compare with full featured with reduced features, the reduced features are providing better accuracy, minimum time and less false positive rate due to feature reduction takes place and removed the unnecessary features.

Consequently, instead of using all 41 characteristics and a different subset of attributes for each layer. Both training and testing performance have increased as a result of these noteworthy outcomes. The performance of our suggested methodology is better in terms of attack detection accuracy. The technique is used to find intrusions based on the KDD dataset. There are numerous assaults in this 41-feature dataset. Forty-one characteristics in all-five for the Probe layer, nine for the DoS layer, fourteen for the R2L layer, and eight for the U2R layer-have been trimmed down [16].

Results provide the F-Value, Accuracy, Precision, and Recall needed to achieve high accuracy. Consequently, it may achieve 100% accuracy for DoS assaults and skew the system [17]. Recall, precision, and F-Value are unaffected by the number of training and test samples.

$$\text{Precision Values} = \frac{T.P.}{T.P. + F.P.} \tag{6}$$

$$\text{Re}call\ \text{Value} = \frac{T.P.}{T.P. + F.N.} \tag{7}$$

$$\text{F Value} = \frac{(1+\beta^2) * \text{Precision Value} * \text{Recall Value}}{\beta^2 * (\text{Precision Value} * \text{Recall Value})} \tag{8}$$

$$\text{Efficiency (\%)} = \frac{\text{Total number of Attacks Detected}}{\text{Total number of Attacks}} * 100 \tag{9}$$

where, T.P., F.P., and F.N. are quantity of True and False Positives and False Negatives, and β correspond to precision versus recall.

5.1 Reduced and 41 features dataset for DoS with 41 features

Table 3 compares two classifiers' outcomes to 41 features and reduced features, both are shown Appendix (A.1) and efficiency are measured. The outcomes are presented in Table 3.

The classification with KDD dataset was completed using 41-feature and 9-feature for DoS. The accuracy of 41 feature datasets is near 98% average for FFNN and PNN. The accuracy of 9 feature is more than 99% average for FFNN and PNN, being the most accurate.

Table 3. Outcomes of DoS with 41 features and reduced features

Function		Features	Efficiency%	Recall	Precision	F-Value
FFNN	Top	**41**	**98.89**	**98.95**	**97.87**	**98.91**
	Normal		96.67	95.77	96.34	97.67
	Poor		94.67	92.31	95.56	94.89
	Top	09	**99.17**	**99.67**	**98.68**	**99.17**
	Normal		99	97.68	98.04	99.01
	Poor		98.33	94.67	96.77	98.36
PNN	Top	**41**	**98.78**	**97.19**	**97.45**	**98.49**
	Normal		98.67	97.07	95.77	98.05
	Poor		96.33	96.67	94.32	97.31
	Top	**09**	**99.24**	**99.45**	**99.74**	**99.28**
	Normal		98.67	99.33	98.54	98.87
	Poor		95.50	99.00	92.77	96.50

Table 4. Outcomes of Probe with 41 features and reduced features

Function		Features	Efficiency%	Recall	Precision	F-Value
FFNN	Top	**41**	94.33	96.33	98.19	94.1
	Normal		93.67	94.35	98.14	92.22
	Poor		91.36	90.86	97.49	90.26
	Top	05	97.67	98.36	99.90	98.89
	Normal		95.16	95.66	98.99	97.26
	Poor		93.33	92.59	96.87	94.67
PNN	Top	**41**	**98.76**	**100**	**98.36**	**98.59**
	Normal		96.60	98.80	98.04	96.01
	Poor		93.67	95.65	97.4	92.68
	Top	**05**	**99.05**	**100**	**99.01**	**99.50**
	Normal		98.17	98.65	98.36	99.17
	Poor		96.5	95.76	97.09	98.52

Table 5. Outcomes of R2L with 41 features and reduced features

Function		Features	Recall	Precision	F-Value
FFNN	Top	**41**	**97.33**	**98.35**	**98.84**
	Normal		95.33	97.71	98.51
	Poor		90.33	97.36	97.84
	Top	14	**99.67**	**99.01**	**99.50**
	Normal		98.45	97.68	96.58
	Poor		93.55	98.68	94.34
PNN	Top	**41**	**96.43**	95.96	93.02
	Normal		95.33	83.33	90.91
	Poor		91.33	81.97	88.09
	Top	**14**	**99.76**	96.96	97.02
	Normal		98.45	86.46	92.74
	Poor		93.55	83.33	90.90

Table 6. Outcomes of U2R with 41 features and reduced features

Function		Features	Efficiency%	Recall	Precision	F-Value
FFNN	Top	**41**	**98.24**	**97.90**	**93.33**	**94.71**
	Normal		96.61	95.11	92.73	93.33
	Poor		94.30	92.63	90.50	91.14
	Top	08	98.89	98.80	96.77	96.78
	Normal		96.30	95.05	93.04	94.78
	Poor		94.98	92.32	91.43	88.46
PNN	Top	**41**	97.98	96.79	99.78	97.27
	Normal		94.67	93.53	98.66	95.05
	Poor		91.73	90.79	96.75	92.09
	Top	**08**	**98.24**	**96.84**	**99.89**	**98.85**
	Normal		97.61	95.32	97.55	95.67
	Poor		94.98	94.79	94.67	90.27

5.2 Reduced and 41 features dataset for probe with 41 features

Table 4 compares two classifiers' outcomes to 41 features and reduced features, both are shown Appendix (A.2) and efficiency are measured. The outcomes are presented in Table 4.

The classification with KDD dataset was completed using 41-feature and 5-feature for DoS. The accuracy of 41 feature datasets is near 96% and 98% average for FFNN and PNN.

The accuracy of 5 feature is more than 98% and 99% average for FFNN and PNN, being the most accurate.

5.3 Reduced and 41 features dataset for R2L

Table 5 compares two classifiers' outcomes to 41 features and reduced features, both are shown Appendix (A.3) and efficiency are measured. The outcomes are presented in Table 5.

The classification with KDD dataset was completed using 41-feature and 5-feature for DoS. The accuracy of 41 feature datasets is near 98% and 94.5% average for FFNN and PNN. The accuracy of 5 feature is more than 99% and 98% average for FFNN and PNN, being the most accurate.

5.4 Reduced and 41 features dataset for U2R with 41 features

Table 6 compares two classifiers' outcomes to 41 features and reduced features, both are shown Appendix (A.4) and efficiency are measured. The outcomes are presented in Table 6.

The classification with KDD dataset was completed using 41-feature and 5-feature for DoS. The accuracy of 41 feature datasets is near 96% and 97% average for FFNN and PNN. The accuracy of 5 feature is more than 97% and 98% average for FFNN and PNN, being the most accurate.

5.5 Comparison of results

The overall results comparison between full featured and reduced features are measured. Table 7 shows the overall results. It is concluded that FFNN is identified for DoS, R2L, and U2L assaults quite effectively and PNN is identified for DoS, The Probe and U2R attacks exhibit a high degree of efficacy. Upon comparison between the full-featured and reduced-featured variants, it has been observed that the latter performs superiorly.

Table 7. A comparison of the performance in different algorithms

	DoS%	**Probe%**	**R2L%**	**U2R%**
Proposed full Features of FFNN	**98.89**	94.33	**98.83**	**98.24**
Proposed reduced Features of FFNN	**99.17**	97.67	**99.50**	**98.89**
Proposed full Features of PNN	**98.78**	**98.76**	96.50	97.98
Proposed reduced Features of PNN	**99.24**	**99.05**	97.56	**98.24**
Layered CRF [1]	97.45	98.65	29.65	86.35
Multi Classifier [16]	97.35	88.73	9.64	29.82
Multi Layer Perceptron [17]	97.24	88.8	5.67	13.27
Gaussian Classifier [8]	82.43	90.4	9.62	22.84

6. CONCLUSION

This research suggests a Neural Network method that makes use of PNN and FFNN classifiers to identify the attacks. This article shows that PNN and FFNN are more accurate for DoS than other approaches. PNN performs more accurately for the Probe than other methods. The FFNN performs better in terms of accuracy than other methods for both R2L and U2R attacks. The KDD dataset is utilized to apply these methods using MATLAB software. For DoS assaults, R2L attacks, and U2R attacks, FFNN performs better than PNN when compared to these two classifiers. As such, it is advised that FFNN approaches be considered in order to increase effectiveness and reduce the false positive rate.

In order to detect network assaults, we suggest using a neural network technique that makes use of PNN and FFNN classifiers. Based on our research, PNN and FFNN perform better than other approaches when it comes to accuracy, especially when it comes to DoS attacks. Especially, PNN outperforms other methods in terms of accuracy against Probe assaults. On the other hand, FFNN is more accurate in R2L and U2R attacks. We use MATLAB software to run these algorithms on the large-scale KDD dataset in order to evaluate our methodology. These two classifiers are compared, and the results show that FFNN performs better than PNN in all attack categories, such as DoS attacks, R2L attacks, and U2R assaults. Therefore, it is advised that FFNN techniques be used in network intrusion detection systems to increase detection effectiveness and lower the False Positive Rate (FPR). To sum up, this study recommends using FFNN classifiers to improve network attack detection accuracy and effectiveness, especially in DoS, R2L, and U2R attack scenarios.

REFERENCES

[1] Devaraju, S. (2019). Evaluation of efficiency for intrusion detection system using Gini index C5 algorithm. International Journal of Engineering and Advanced Technology, 8(6): 2196-2200. https://doi.org/10.35940/ijeat.F8593.088619

[2] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A. (2009). A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, pp. 1-6. https://doi.org/10.1109/CISDA.2009.5356528

[3] Overman, E. (2011). A MATLAB tutorial. Department of Mathematics, Ohio State University, Columbus, OH, 180.

[4] Devaraju, S., Ramakrishnan, S. (2014). Performance comparison for intrusion detection system using neural network with KDD dataset. ICTACT Journal on Soft Computing, 4(3): 743-752. https://doi.org/10.21917/ijsc.2014.0106

[5] Eesa, A.S., Orman, Z., Brifcani, A.M.A. (2015). A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. Expert Systems with Applications, 42(5): 2670-2679. https://doi.org/10.1016/j.eswa.2014.11.009

[6] Soni, D., Tiwari, V., Kaur, B., Kumar, M. (2021). Cloud computing security analysis based on RC6, AES and RSA algorithms in user-cloud environment. In 2021 First International Conference on Advances in Computing and Future Communication Technologies (ICACFCT), Meerut, India, pp. 269-273. https://doi.org/10.1109/ICACFCT53978.2021.9837360

[7] Lin, W.C., Ke, S.W., Tsai, C.F. (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. Knowledge-Based Systems, 78: 13-21. https://doi.org/10.1016/j.knosys.2015.01.009

[8] Ramakrishnan, S., Devaraju, S. (2017). Attack's feature selection-based network intrusion detection system using fuzzy control language. International Journal of Fuzzy Systems, 19(2): 316-328. https://doi.org/10.1007/s40815-016-0160-6
[9] Sharma, V., Soni, D., Srivastava, D., Kumar, P. (2021). A novel hybrid approach of suppression and randomization for privacy preserving data mining. Ilkogretim Online, 20(5): 2451-2457. https://doi.org/10.17051/ilkonline.2021.05.267
[10] Devaraju, S., Prakash, D.S. (2019). Developing efficient web-Based XML tool. International Journal of Recent Technology and Engineering, 8(3): 8580-8584.
[11] Soni, D., Kumar, M. (2017). Secure data communication in client-cloud environment: A survey. In 2017 7th International Conference on Communication Systems and Network Technologies (CSNT), Nagpur, India, pp. 246-252. https://doi.org/10.1109/CSNT.2017.8418546
[12] Jawahar, S., Harishchander, A., Devaraju, S., Reshmi, S., Manivasagan, C., Sumathi, P. (2020). Efficiently mining closed sequence patterns in DNA without candidate genertion. International Journal of Life science and Pharma Research, SP-08: 14-18.
[13] Soni, D., Sharma, V., Srivastava, D. (2019). Optimization of security issues in adoption of cloud ecosystem. In 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, pp. 1-5. https://doi.org/10.1109/IoT-SIU.2019.8777670
[14] Soni, D., Kumar, M. (2019). An automated cloud security framework based on FCM in user-cloud environment. International Journal of Engineering and Advanced Technology, 8(6): 3235-3240. https://doi.org/10.35940/ijeat.F8831.088619
[15] Soni, D., Srivastava, D., Bhatt, A., Aggarwal, A., Kumar, S., Shah, M.A. (2022). An empirical client cloud environment to secure data communication with alert protocol. Mathematical Problems in Engineering, 2022(1): 4696649. https://doi.org/10.1155/2022/4696649
[16] Sellappan, D., Srinivasan, R. (2021). Fuzzy rule-based layered classifier and entropy-Based feature selection for intrusion detection system. In Handbook of Research on Cyber Crime and Information Privacy. IGI Global. IGI Global, 2: 289-309. https://www.igi-global.com/gateway/chapter/261735.
[17] Devaraju, S., Ramakrishnan, S., Jawahar, S., Soni, D., Somasundaram, A. (2022). Entropy-based feature selection for network intrusion detection systems. In Methods, Implementation, and Application of Cyber Security Intelligence and Analytics. IGI Global, pp. 201-225. https://doi.org/10.4018/978-1-6684-3991-3.ch012
[18] Gupta, K.K., Nath, B., Kotagiri, R. (2008). Layered approach using conditional random fields for intrusion detection. IEEE Transactions on dependable and secure Computing, 7(1): 35-49. https://doi.org/10.1109/TDSC.2008.20
[19] Sheikhan, M., Jadidi, Z., Farrokhi, A. (2012). Intrusion detection using reduced-size RNN based on feature grouping. Neural Computing and Applications, 21: 1185-1190. https://doi.org/10.1007/s00521-010-0487-0
[20] Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity, 2(1): 1-22. https://doi.org/10.1186/s42400-019-0038-7
[21] Hoque, M.S., Mukit, M.A., Bikas, M.A.N. (2012). An implementation of intrusion detection system using genetic algorithm. International Journal of Network Security & Its Applications (IJNSA), 4(2): 109-120. https://doi.org/10.5121/ijnsa.2012.4208
[22] Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A., Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7: 41525-41550. https://doi.org/10.1109/ACCESS.2019.2895334
[23] Kumar, M., Hanumanthappa, M., Kumar, T.S. (2012). Intrusion Detection System using decision tree algorithm. In 2012 IEEE 14th International Conference on Communication Technology, Chengdu, China, pp. 629-634. https://doi.org/10.1109/ICCT.2012.6511281

APPENDIX

A Feature Selection

A.1 Features selected for DoS layer

Feature Number	Feature Name
1	Duration
2	protocol_type
4	Flag
5	src_bytes
23	Count
34	dst_host_same_srv_rate
38	dst_host_serror_rate
39	dst_host_srv_serror_rate
40	dst_host_rerror_rate

A.2 Features selected for Probe layer

Feature Number	Feature Name
1	Duration
2	protocol_type
3	Service
4	Flag
5	src_bytes

A.3 Features selected for R2L layer

Feature Number	Feature Name
1	Duration
2	protocol_type
3	Service
4	Flag
5	src_bytes
10	Hot
11	num_failed_logins
12	logged_in
13	num_compromised
17	num_file_creations
18	num_shells
19	num_access_files
21	is_host_login
22	is_guest_login