ACCURACY DETECTION OF NETWORK INTRUSION DETECTION SYSTEM USING NEURAL NETWORK CLASSIFIER ON THE KDD DATASET

A. Somasundaram[1], S. Devaraju[2], M.Thenmozhi[3] and S. Jawahar[4]

[1] Department of Computer Science and Applications, Sri Krishna Arts and Science College, Coimbatore, Tamil Nadu, India

[2] School of Computing Science and Engineering, VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh, India.

[3] Department of Artificial Intelligence and Data Science, Sri Eshwar College of Engineering, Coimbatore, Tamil Nadu, India

[4] School of Sciences, Christ Deemed-to-be University, Delhi NCR, India

[1]Corresponding Author Email:

devamcet@gmail.com

Abstract – Network Intrusion Detection System, Security mechanism has recently become vital component of IT world and system usage over the internet. As an effective technique to dealing with network difficulties, intrusion detection systems employ several classifiers to detect various types of attacks. The performance results are compared with several classifiers of neural network. Classifiers are employed with different five classifiers in this proposed study namely Elman Neural Network (ENN), Feed Forward Neural Network (FFNN), Probabilistic Neural Network (PNN), Radial Basis Neural Network (RBNN) and Generalized Regression Neural Network (GRNN). The feature decline approaches used to filter the specific KDD dataset in this problem. The accuracy results of full-features and reduced features datasets are matched.

Keywords - Neural networks, Intrusion detection, ENN, FFNN, PNN, RBNN, GRNN, MATLAB, KDD Dataset.

1. INTRODUCTION

Network Intrusion Detection System (NIDS), there have been few intruders in recent years, so the user may easily control them either known nor unknown threats. In recent centuries, the security has emerged as most critical concern in standings of safeguarding valuable information. Since intruders are presenting novel types incursion into the flea market, user is unable to govern their system.

The attacks are divided to two types: signature and anomaly. The signature identifies intrusions compare with their constraints to the signatures already in the database. It's an incursion if assaults are matched with the signatures. Signature-based intrusions are referred to be known assaults since users discover the intrusion by matching signature logs. The logs store variety of known assaults which are detected from networks. Unknown attacks are anomaly attacks that are detected on the network because they deviate from conventional attacks.

NIDS distinguish between network and host related threats and anomaly assaults are available [1]. The interconnectedness of computer systems detects network-based assaults. When the systems communicate with one another, attack is transmitted through routers and switches while connecting the system each other's. Single system is connected to detect the assaults and are simple to prevent. The assaults are mostly carried out by externally linked devices. Pen drives, CDs,

VCDs, and floppy discs are examples of external devices. Web assaults are conceived while interconnecting systems via internet. These assaults are distributed across multiple computers.

The various neural network classifiers are proposed in this paper to detect signature-based infiltration. MATLAB application is used to address this problem utilizing several strategies for enhancing performance on the KDD dataset. Compare and analyze the full and reduced features. The Figure 1 indicates the taxonomy of proposed systems:

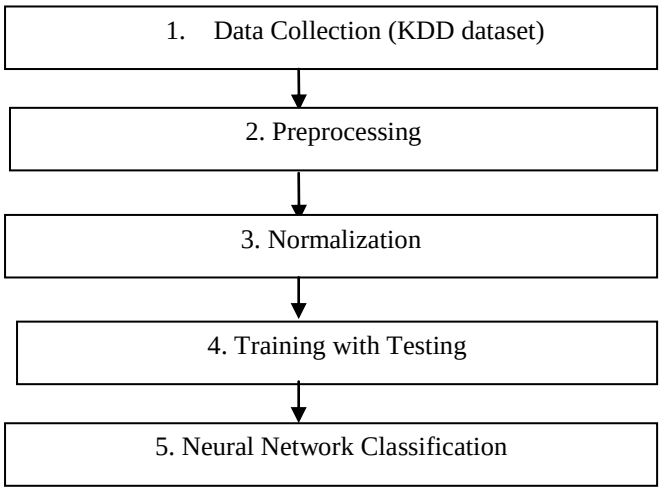

Fig. 1: Intrusion Detection ladders

Following is the remainder of this paper: Literature review utilized in Section-2. Section-3 labels KDD Cup dataset. Section-4 deliberates proposed methodologies. Section-5 offerings experimental data and discussion and describes the conclusion in Section-6.

2. LITERATURE REVIEW

In NIDS for past 30 years, assaults detection systems in evolving. Many approaches and strategies have been developed, and many systems have been impacted by various invasions. Data mining, neural networks, and statistical algorithms are among the techniques used to detect intrusions [2]. The many strategies and tactics are explained in this linked paper.

The two challenges are Accuracy and Efficiency, which are handled by layered with Conditional Random CRF). In this technique displays great attack detection accuracy and efficiency. For detecting attacks, this method employs the KDD dataset intrusion detection data set [1]. Rough Set NN technique applied to decreases amount of resources needed to identify the assault. To test the dataset to get more reliable results [3][4]. Anomaly detection is determined using Multivariate Statistical Analysis techniques. The statistical technique is applied to relate the performance [7][8].

Data mining methods are decision trees which are recycled to detect assaults. Train and test the KDD dataset. In this model performs better to detecting novel kinds of anomalies [9][10]. The HMM model is applied to demonstrate the anomaly intrusion detection depends on system calls [11][12]. HMM approaches that can detect and classify accurately for numerous anomaly behaviors. For example scanning the network helps the system to identify the threats and DDoS attacks are used to distribute the threats over network using matrix of Correlation Coefficient [13][14]. The HMM is applied to demonstrate and control anomaly intrusion detection built on system calls [15][16][17].

Using statistical classification techniques, the anomalies are detected using the model of Hierarchical Gaussian Mixture. The well-known KDD99 dataset is used to evaluate this model. Six categorization strategies are utilized to validate the feasibility and effectiveness. This approach is utilized in Intrusion Detection Systems [18][19] to reduce false alarms and improve attack accuracy.

The Genetic Algorithm(GA) is recycled to identify intrusions. Using GA to encode the problem, it takes into account information with temporal as well as spatial. Genetic Algorithm is more useful for detecting network anomalies [20][21][22]. To lower the computational intensity, several feature bargain are applied. KDD dataset is utilized to minimize processing calculation time and increase accuracy of the system [23][24][25].

The various strategies are examined using various criteria. The suggested system takes into account the shortcomings of the present system and proposes to fix the issues with the KDD dataset using a neural network classifier.

3. DESCRIPTION ABOUT KDD DATASET

KDD dataset was APPLIED to estimate anomaly detection algorithms. Training dataset comprises approximately 4,900 K single association vectors, each connection 41 features comprising, considered either attack or normal. Four categories of attacks [5]. There are 24 training attack kinds in the datasets and additionally 14 types are available in test data.

3.1 Data Collection

buffer_overflow, ftp_write, imap, Back, guess_passwd, land, ipsweep, loadmodule, neptune, multihop, perl, phf, teardrop, nmap, pod, rootkit, portsweep, spy, smurf, normal, satan, warezmaster, warezclient are all attacks in KDD dataset. Four types of classes are derived from various attacks[5][13]. Table 1 indicates collection of attacks type:

Table 1. Attacks Types

DoS (Denial of Services)	R2L (Remote-to-Local)	U2R (User-to-Root)	Probe
teardrop back land pod Neptune smurf	ftp_write guess_passwd imap phf multihop spy warezmaster warezclient	buffer_overflow Loadmodule Perl rootkit	portsweep nmap satan ipsweep

DoS is deny legitimate system requests, such as flooding, U2R is unlicensed access root privileges, such as different buffer overflow, R2Lis unlicensed access from remote system, such as password guessing; and Probing: surveillance as well as other probing, such as scanning port [7].

Labelled the sets are A1, B1, C1, D1, and E1, in that order. Set 'A1' obtains data starting the DoS. Set 'B1' collects data starting the U2R. Set 'C1' obtains data starting the R2L. Set 'D1' collects data starting the Probe. Set 'E1' obtains data starting the Normal. The subsequent data sets applied to train and evaluate the given dataset. Dataset is considered using 41 features and 13 reduced features [5].

Table 2. Dataset Collection Training and also Testing

	10% dataset is used for 41 features and 13 features for classification
DoS	3,91,458
U2R	52
R2L	1,126
Probe	4,107
Normal	97,278
Total	**4,94,021**

3.2 Preprocessing

The neural network's input data must be in the range [0 1] or [-1 1]. As a result, data preparation and normalization are necessary. KDD format data has been preprocessed. Each KDD record comprises 41 attributes namely discrete, continuous, or symbolic in widely variable ranges.

Each symbol is assigned an integer code for conversion into numerical form. Protocol type feature, apply numeric value for each protocol. The titles of attacks are initially assigned to any one classes: 'A1' is for DoS, 'B1' is for U2R, 'C1' is for R2L, 'D1' is for Probe, and 'E1' is for Normal. The src_bytes value ranges between 0 and 1.3 billion and similarly for dst_bytes value ranges between 0 and 1.3 billion with extremely large integer range. These attributes are subjected to logarithmic scaling (with base 10) to narrow the range between 0.0 and 9.14. The remaining features are Boolean, with values between 0.0 and 1.0.

3.3 Normalization

Statistical examination is conducted to each feature with the range of values to identify with their characteristic with maximum value. Range [0,1] for feature values are interpreted for normalization using maximum values and subsequent formula as If (feature_value > maximum_feature) normalized_value=1; Otherwise normalized_value = (feature_value / maximum_feature)

4. PROPOSED METHODOLOGIES

4.1. Radial Basis Neural Network (RBNN)

RBNN includes three layers: (1) input, (2) hidden and (3) output layer. Neurons of hidden layer have functions of Gaussian transfer with outputs from the neuron's centre distance that inversely proportional. Figure 2 represents the structure. It is considered of curve-fitting issue for high-dimension. It is experiment weight for every neuron to find distance.

$$\text{Weight\_Value} = \text{RBF}\left(\text{Distance\_Value}\right) \quad (1)$$

Training method determines the following parameters:

1. Buried layer contains the numerous neurons.
2. RBF function's centre coordinates by hidden-layer.
3. Dimensions of RBF function by radius.
4. RBF weights function is applied for outputs as they pass through summing layer.

RBF approaches were utilized to train the networks. K-means-clustering is utilized to locate the centres of cluster, these are utilised in centre of functions, with training points random subset serving as the centres.

4.2 Generalized Regression Neural Network (GRNN)

When goal variable is continuous, Regression is carried out via General Regression Neural Networks. GRNN network selection, DTREG select appropriate network based on the target variable level. When compared to Multilayer Perceptron networks, GRNN networks offer the following advantages and disadvantages:

- Training a GRNN network is typically significantly faster than training network of multilayer.
- Networks of GRNN frequently outperform multilayer perceptron networks in terms of accuracy.
- GRNN networks are mostly unaffected by outliers.
- When it comes to classifying new cases, multilayer perceptron networks are faster than GRNN networks.
- GRNN demand greater memory space.

GRNN networks are composed of four layers:

1. In the input layer, one neuron is assigned to each predictor variable. When dealing with definite variables, in which neurons of 'n-1' are employed, where 'n' is definite number. These values are fed every neuron in the layer of hidden by the input neurons.
2. Hidden layer – for training, one neuron is assigned to each case. Along with the target value, neuron retains values of every predictor variable. Pattern layer receives neurons from resulting value.
3. Pattern / Summation layer —It has two levels of neurons. Summation unit is one neuron in denominator and others in numerator.

4.3 Feed Forward Neural Network (FFNN)

FFNN receive signals and convert from input into output signal. FFNNs are typically modest networks which link inputs into outputs and active in pattern recognition. Single and Multi are two types of FFNN.

Single layer is the first basic learning machine. The term "single layer" refers to having two layers, namely input and output. The term "multi layer" refers to having three layers namely input, hidden, output layer.

In multi-layer, two sorts of phases. Forward level is applied to set free parameter and ends with computing error signal.

$$\text{error\_signal}_i = \text{desired}_i - \text{actual}_i \quad (2)$$

Where, input is desired response and actual output in response. Error_signal is spread during Backward to minimize the error need to adjust the parameters. in statistical way.

4.4 Probabilistic Neural Network (PNN)

It is natural extension from classifiers of Bayes. To be specific, it is viewed as function that estimates the possibility density. It is made by nodes that are organized into three layers of input namely pattern, summation and layer of output.

A. Pattern Layer: Phases of each training has one node pattern. It is product weight, which weights enter the node from specific node. Following that, product goes over the activation function:

$$\text{expression}\left[\left(\mathbf{x}^{\mathrm{T}}\mathbf{w}_{ki} - 1\right) / \sigma^2 \right] \quad (3)$$

B. Summation Layer: To get output pattern from each summation nodes belonging to specific class:

$$\Sigma_{i=1}^{Nk}\text{expression}\left[\left(\mathbf{x}^{\mathrm{T}}\mathbf{w}_{ki} - 1\right) / \sigma^2 \right] \quad (4)$$

C. Output Layer: Categorization decision is made by binary neurons at the output nodes.

$$\Sigma_{i=1}^{Nk}\text{expression}\left[\left(\mathbf{x}^{\mathrm{T}}\mathbf{w}_{ki} - 1\right) / \sigma^2 \right] > \Sigma_{i=1}^{Nj}\text{expression}\left[\left(\mathbf{x}^{\mathrm{T}}\mathbf{w}_{kj} - 1\right) / \sigma^2 \right] \quad (5)$$

The single factor that must be chosen for training:

- too tiny deviations result is too tiny in pointed approximation;
- deviations are too big in smooth out details.

4.5 Elman Neural Network (ENN)

These networks are also called as feedforward with tap interruptions on recurrent links layer. It has three-layered, including "context units" input. The buried layer has weighted connections to these context units. At every iteration, input is disseminated using regular feed-forward method, after learning rule implemented. Context units preserve replica hidden units' prior because fixed back connections propagate when rule is useful.

Hidden unit activate both input and context units, which activate the output. Concealed entities also trigger the context entities. It is referred to as forward activation. Depending on the work, this time cycle may or may not involve a learning phase. If this is the case, compare the output into input. The value for recurrent setups is set to 1.0 it cannot be modified. At 't+1' step, repeat the preceding sequence. Context units include values that closely match concealed time 't' from unit values. Following happens while using the function to train Elman network. At every step:

1. The network is fed the whole input sequence, output is measured and yield sequence error.
2. Back propagated error for each step discover gradients of errors in every bias and weight. Because contributions weights and biases also mistakes via disregarded of recurrent connection, it is an approximation.
3. The gradient is utilized to update weights with user-selected back prop function with training. Function traingdx is suggested.

5. RESULTS AND DISCUSSION

Based on the KDD dataset, IDS algorithms utilized to detect malicious. This dataset includes 41 features and reduced features. Using the PNN, accuracy was enhanced by 96.23% by decreasing 41 characteristics to 13. These datasets may be used using the MATLAB software [6], and when compared to the other five NN classifiers, PNN has highest accurateness [2].

Attack Detection Rate (ADR): Overall attacks detected ratio and entire attacks.

$$A.D.R.=\frac{(Tr.\ Pos.\ +\ Tr.\ Neg.)}{(Tr.\ Pos.\ +\ Fal.\ Pos.\ +\ Fal.\ Neg.\ +\ Tr.\ Neg.)}*100 \qquad (6)$$

FPR: Entire number of misclassified ratio occurrences and entire normal occurrences.

$$F.P.R.\ =\ \frac{Tot.\ no.\ of\ Misclassified\ Instances)}{Tot.\ Normal\ Instances}*100 \qquad (7)$$

5.1 Experimental Results for Total Features

Below table shows the five classes with five classifiers also utilized, and efficiency is measured. Classification of 41 highlighted datasets presented in Table 3.

Table 3: Experimental Outcomes for Total Features

Classes/ Networks	DoS	U2R	R2L	Probe	Normal	Efficiency (%)	False Positive Rate	Time Taken (s)
FFNN	376281	43	719	3846	89722	95.26	2.47	
ENN	372814	45	719	3641	88798	94.33	2.89	
GRNN	365818	39	703	3587	85327	92.20	4.02	127
PNN	390964	48	943	3876	91148	98.57	1.3	
RBNN	370946	47	861	3789	89421	94.14	1.87	

Based on these findings, a graphical representation is provided in the chart below. Figure 3 depicts the detection rate results for 41 features.

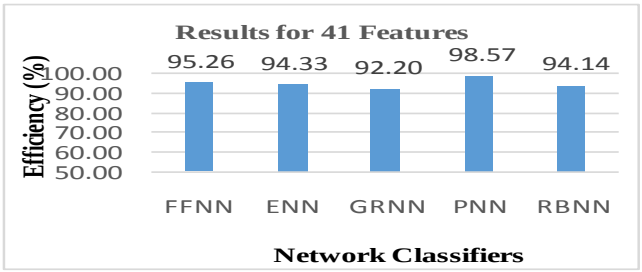

Fig. 3: Efficiency for full features with classifiers

The categorization was performed with 41-feature dataset. Table 3 shows the accuracy percentages for five neural networks. The accuracy of FFNN is 95.26%, ENN is 94.33%, GRNN is 92.20%, PNN is 98.57%, and RBN is 94.14%. Figure 4 depicts the FPR results for 41 features.

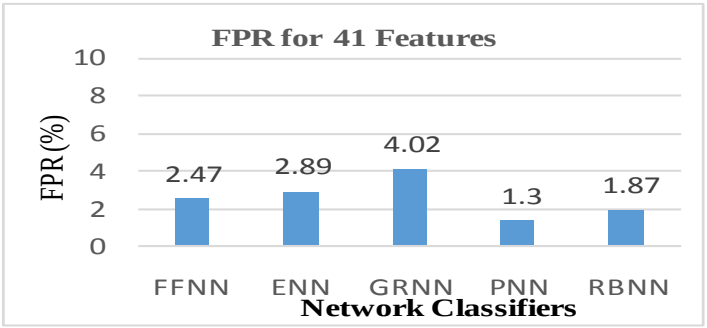

Fig. 4: FPR for full features with classifiers

The categorization was performed with 41-feature dataset. Table 4 shows the FPR percentages for five neural networks. The FPR of FFNN is 2.47%, ENN is 2.89%, GRNN is 4.02%, PNN is 1.3%, and RBN is 1.87%. Minimum time is take for processing.

5.2 Experimental Results for Reduced Features

Dimensionality drop techniques for compression and data analysis is Principal Component Analysis(PCA). It is a valuable analysis tool because patterns might be difficult to uncover in high-dimensional data. When patterns are identified, it will be compressed to reduce the sum of dimensions without information damage [6]. In data Given, if each datum contains 'N' characteristics and formula for average observation is:

$$\mu = \frac{1}{num}\sum_{i=1}^{num} x_i \qquad (8)$$

The standard deviation is well-defined as

$$\Phi_i = X_i - \mu \qquad (9)$$

Top 13 features are represented in table 4 using PCA.

Table 4: Top 13 Features after reduction

Reduced 13 Features Dataset	
001- duration 001- flag 002- src_bytes 003- dst_bytes 004- land 005- wrong_fragment 006- urgent 007- num_failed_logins	008- logged_in 009- dst_host_serror_rate 010- dst_host_srv_serror_rate 011- dst_host_rerror_rate 012- dst_host_srv_rerror_rate

Once selected 13 features, utilized the dataset to categories the neural networks. The classification of 13 highlighted datasets is presented in Table 5.

Table 5: Outcomes for 13 Features Dataset

Classes/ Networks	**DoS**	**U2R**	**R2L**	**Probe**	**Normal**	**Efficiency (%)**	**False Positive Rate**	**Time Taken (s)**
FFNN	387663	45	782	3923	90987	97.85	1.78	103
ENN	380814	45	719	3641	88798	95.95	2.09	
GRNN	379526	42	839	3773	90421	96.07	2.67	
PNN	390913	49	974	3911	93214	99.00	0.92	
RBNN	373442	45	8591	3547	85477	95.36	1.49	

Based on findings, a graphical representation is provided. In figure 5 depicts the detection rate results for 13 features.

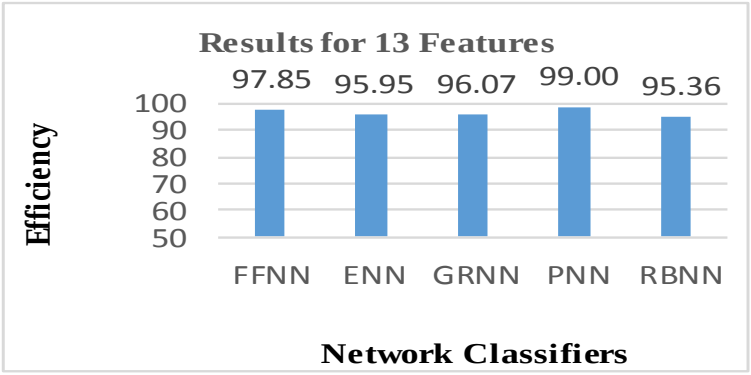

Fig. 5: Accuracy (%) for 13 features

Categorization was performed with 13-feature dataset. Table 5 shows the accuracy percentages for five neural networks. The accuracy of FFNN is 97.85%, ENN is 95.95%, GRNN is 96.07%, PNN is 99%, and RBN is 95.36%. Figure 6 depicts the FPR results for 41 features.

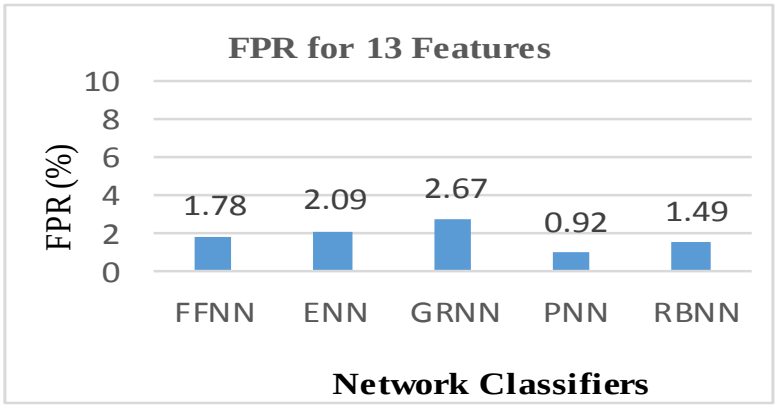

Fig. 6: FPR (%) for 13 selected features

The categorization was performed with 13-feature dataset. Table 5 shows the FPR percentages for five neural networks. The FPR of FFNN is 1.78%, ENN is 2.09%, GRNN is 2.67%, PNN is 0.92%, and RBN is 1.49%. Minimum time is take for processing.

6. SUMMARY

In this article, there are five types of neural network classifiers are applied to categorize the detection rate, FPR and time taken. Benchmarking KDD dataset is applied in this study for experiments. It is observed that reduced feature is outperforming superior than full features. Detection rate is somewhat increased for all five neural network classifiers based on the comparison between full feature detection rate and reduced feature detection rate. Similarly, the FPR is outperforming better than the full features. Also reduce the time take in the reduced features. Based on this comparison, the reduced features are outperforming better than full features in terms of improving ADR, lessens the FPR and minimizes time taken.

7. CONCLUSION

This research proposes innovative approach to detect intrusions by classifiers. This results shows that PNN outperform the ENN, FFNN, RBN and GRNN in terms of accuracy. The feature reduction strategies are applied to improve the outcomes. The PCA is applied to decrease the characteristics of the dataset and experimented with MATLAB Tool. Total of 13 best features are selected by PCA from 41 features. The 13 features are fed into five neural network classifiers, after that the results are matched. Overall results reveal that 13 features are more efficient than 41 features, applying with shorter testing and training timeframes. When matching five classifiers, PNN outperforms ENN, FFNN, RBN and GRNN. The PCA-reduced dataset yields encouraging results. As a result, it is suggested that we pursue feature reduction strategies in our future research to enhance ADR and lower the FPR. In future, ML and deep learning algorithms will apply to achieve more detection rate and less FPR.

REFERENCES

1. Gupta, K. K., et al., R., (2008). Layered approach using conditional random fields for intrusion detection, IEEE Transactions on dependable and secure Computing, vol 7, iss 1, pp. 35-49.
2. Devaraju S. & Ramakrishnan S., (2013). Performance Comparison of Intrusion Detection System using Various Techniques – A Review, ICTACT Journal on Communication Technology, vol 4, iss 3, pp.802-812.
3. Devaraju S., Ramakrishnan S., (2011). Performance Analysis of Intrusion Detection System Using Various Neural Network Classifiers, IEEE International Conference on Recent Trends in Information Technology (ICRTIT 2011), pp.3-5.
4. Gang Wang, et al. (2010). A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering, Elsevier Expert Sys. with Appl., vol 37, pp.6225–6232.
5. KDD Intrusion Detection Data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 2010.
6. MATLAB (MATrix Laboratory) tutorials, http://terpconnect.umd.edu/~nsw/ench250/matlab.htm
7. Devaraju S. & Ramakrishnan S., (2014). Performance Comparison for Intrusion Detection System using Neural Network with KDD Dataset, ICTACT Journal on Soft Computing, vol 4, iss 3, pp.743-752.
8. Devaraju S. & Ramakrishnan S., (2013). Detection of Accuracy for Intrusion Detection System using Neural Network Classifier, International Journal of Emerging Technology and Advanced Engineering, vol 3, iss 1, pp.338-345.
9. Nadiammai GV, Hemalatha M, (2014). Effective Approach toward Intrusion Detection System using Data Mining Techniques, Elsevier Egyptian Informatics Journal, vol 15, pp.37-50.
10. Minjie Wang, Anqing Zhao, (2012). Investigations of Intrusion Detection Based on Data Mining, Springer Recent Advances in Computer Science and Information Engineering Lecture Notes in Electrical Engineering, vol 124, pp.275-279.
11. Shingo Mabu, et al., (2011). An Intrusion-Detection Model Based on Fuzzy Class-Association-Rule Mining Using Genetic Network Programming, IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, vol 41, iss 1, pp.130-139.
12. Adel SabryEesa, et al., (2015). A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems, Elsevier Expert Systems with Applications, vol 42, pp.2670–2679.
13. Devaraju S. &Ramakrishnan S., (2015). Detection of Attacks for IDS using Association Rule Mining Algorithm, IETE Journal of Research, vol 61, iss 6, pp.624-633.
14. Wei-Chao Lin, et al., (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors, Elsevier Knowledge-Based Systems, vol 78, pp.13–21.
15. Ramakrishnan S. & Devaraju S., (2017). Attack's Feature Selection-Based Network Intrusion Detection System Using Fuzzy Control Language, Springer-International Journal of Fuzzy Systems, vol 19, iss 2, pp.316-328.
16. Devaraju S., (2019). Evaluation of Efficiency for Intrusion etection System Using Gini Index C5 Algorithm, International Journal of Engineering and Advanced Technology (IJEAT), vol 8, iss 6, pp.2196-2200.
17. Suseela T. et al., (2005). Hierarchical Kohonenen Net for Anomaly Detection in Network Security, IEEE Tr. on Sys. Man and Cybernetics, vol 35, iss 2, pp.302-312.
18. Devaraju S. & SaravanaPrakash D., (2019). Developing Efficient Web-Based XML Tool, International Journal of Recent Technology and Engineering (IJRTE), vol 8, iss 3, pp.8580-8584.
19. Devaraju S. & SaravanaPrakash D., (2019). Total Benefit Administration for Industry Environment, TEST Engineering and Management, vol 81, pp.4594-4599.
20. Shih-Wei Lin, et al., (2012). An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection, Elsevier Applied Soft Compu., vol 12, iss 10, pp.3285-3290.
21. Jawahar S., et al., (2020). Efficiently Mining Closed Sequence Patterns in DNA without Candidate Genertion, International Journal of Life science and Pharma Research (Special issue on Advancements in Applications of Microbiology and Bioinformatics Inpharmacology), SP-08, iss 8, pp.14-18.
22. Gaik-Yee Chan, et al., (2013). Discovering fuzzy association rule patterns and increasing sensitivity analysis of XML-related attacks, Elsevier Jr. of Netw. and Compu. Appl., vol 36, pp.829–842.
23. Devaraju S and Ramakrishnan S, (2019). Association Rule-Mining-Based Intrusion Detection System with Entropy-Based Feature Selection: Intrusion Detection System (Chapter 1), Handbook of Research on Intelligent Data Processing and Information Security Systems, IGI Global, DOI: 10.4018/978-1-7998-1290-6, pp. 1-24, Pages: 24.
24. Devaraju S and Ramakrishnan S, (2020). Fuzzy Rule-Based Layered Classifier and Entropy-Based Feature Selection for Intrusion Detection System (Chapter 15), Handbook of Research on Cyber Crime and Information Privacy (2 Volumes), IGI Global, DOI: 10.4018/978-1-7998-5728-0, Pages: 753.
25. Devaraju, S., et al., (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.