ISSN: 2229-6956(ONLINE)
DOI: 10.21917/ijsc.2014.0106

PERFORMANCE COMPARISON FOR INTRUSION DETECTION SYSTEM USING NEURAL NETWORK WITH KDD DATASET

S. Devaraju[1] and S. Ramakrishnan[2]
[1]*Department of Computer Applications, Dr. Mahalingam College of Engineering and Technology, India*
E-mail: deva_sel@yahoo.com
[2]*Department of Information Technology, Dr. Mahalingam College of Engineering and Technology, India*
E-mail: ram_f77@yahoo.com

Abstract
Intrusion Detection Systems are challenging task for finding the user as normal user or attack user in any organizational information systems or IT Industry. The Intrusion Detection System is an effective method to deal with the kinds of problem in networks. Different classifiers are used to detect the different kinds of attacks in networks. In this paper, the performance of intrusion detection is compared with various neural network classifiers. In the proposed research the four types of classifiers used are Feed Forward Neural Network (FFNN), Generalized Regression Neural Network (GRNN), Probabilistic Neural Network (PNN) and Radial Basis Neural Network (RBNN). The performance of the full featured KDD Cup 1999 dataset is compared with that of the reduced featured KDD Cup 1999 dataset. The MATLAB software is used to train and test the dataset and the efficiency and False Alarm Rate is measured. It is proved that the reduced dataset is performing better than the full featured dataset.

***Keywords*:**
Intrusion Detection, Neural Networks, KDD Cup 1999 Dataset, MATLAB

1. INTRODUCTION

Over the Internet, the users are sharing their valuable information all over the world. Internet has also created numerous ways to compromise the stability and security of the systems connected with each other. The two kinds of mechanisms are static and dynamic. The static mechanisms such as firewalls and software updates provide a reasonable level of security and dynamic mechanisms such as intrusion detection systems. In the previous century, there was less number of intruders so the user can manage them easily from the known or unknown attacks. In present years the security is the most serious issue for securing the valuable information. Therefore either static mechanism or dynamic mechanism is required for protecting individual information despite the prevention techniques. The intrusion detection system is useful not only in detecting successful intrusions, but also in monitoring or preventing the attacks for timely countermeasures [1].

Intrusion detection attacks can be classified into two groups: Misuse or Signature based and Anomaly based Intrusion Detection. The misuse or signature based intrusion detection system detects the intrusion by comparing with its existing signatures in the database. If the detecting attacks and signatures match, it is an intrusion. The signature based intrusions are called known attacks whenever the users are detecting the intrusion by matching with the signatures log files. The log file contains the list of known attacks detected from the computer system or networks. The anomaly based intrusion detection is called as unknown attacks and this attack is observed from network as it deviates from the normal attacks.

The intrusion detection systems are classified as Network based or Host based attacks. The network based attacks may be either misuse or anomaly based attacks. The network based attacks are detected from the interconnection of computer systems. Since the system communicates with each other, the attack is sent from one computer system to another computer system by the way of routers and switches. The host based attacks are detected only from a single computer system and is easy to prevent the attacks. These attacks mainly occur from some external devices which are connected. The web based attacks are possible when systems are connected over the internet and the attacks can be spread into different systems through the email, chatting, downloading the materials etc. Nowadays many computer systems are affected from web based dangerous attacks.

In this system, it is proposed to detect signature based intrusion using neural network classifier Feed Forward Neural Network (FFNN), Generalized Regression Neural Network (GRNN), Probabilistic Neural Network (PNN) and Radial Basis Neural Network (RBNN). The various techniques are applied in this problem in MATLAB application for improving the best performance applied to KDD Cup 1999 dataset. The performance of the full featured dataset and reduced dataset is analyzed.

The remaining of this paper is given as follows: In section 2, the related work used for intrusion detection is discussed. Section 3 discusses Feed Forward Neural Network, section 4 discusses Generalized Regression Neural Network, section 5 discusses Probabilistic Neural Network and section 6 discusses Radial Basis Neural Network, section 7 describes about the KDD Cup dataset Description. Section 8 gives our experimental results and discussion and section 9 deals with conclusion.

2. RELATED WORK

The intrusion detection system has a critical role in detecting the intrusion in the real world. A number of methods and techniques have been proposed as many systems have been affected by a variety of intrusions. The various techniques used to detect the intrusions are data mining, neural network and statistical methods. In this related work, the various methods and techniques for detecting intrusion detection systems are discussed.

The Multivariate Statistical Analysis methods are used to determine the anomaly detection. The statistical methods are used to compare the performance of the system [2]. The Hidden Markov Model is used to implement and determine the system call based anomaly intrusion detection [3] and [4].

Conditional Random Fields and Layered Approach are addressed by the two issues of Accuracy and Efficiency. This approach demonstrates the high attack detection accuracy and high efficiency using Conditional Random Fields and Layered Approach. This approach uses KDD Cup '99 intrusion detection data set for detecting the attacks [1].

Recurrent Neural Network model used with four groups of input features has been proposed as misuse-based IDS and the experimental results have shown that the reduced-size neural classifier has improved classification rates, especially for R2L attack [5].

The Genetic Algorithm is used to detect the intrusions in networks. It considers both temporal and spatial information of network connections during the encoding of the problem using Genetic Algorithm. The Genetic Algorithm is more helpful for identification of network anomalous behaviors [6] and [7]. The Rough Set Neural Network Algorithm is used to reduce a number of computer resources required to detect an attack. The KDD Cup'99 dataset is used to test the data and gives the better and robust result [8]. The various feature reduction techniques such as Independent Component Analysis, Linear Discriminant Analysis and Principal Component Analysis are used to reduce the computational intensity. KDD Cup 99 dataset is used to reduce computation time and improve the accuracy of the systems [9].

The Hierarchical Gaussian Mixture Model detects network based attacks as anomalies using statistical classification techniques. This model is evaluated by well known KDD99 dataset. There are six classification techniques used to verify the feasibility and effectiveness. This technique is used to reduce the missing alarm and accuracy of the attack in Intrusion Detection System [10].

Anomaly detection and analysis are based on the methods which describe the normal and abnormal traffic and accurately detect and classify various anomaly behaviors based on Correlation Coefficient Matrix [11]. The data mining techniques like decision trees are used to detect the attacks. The KDD 99 dataset is used for training and testing the data. This model has shown improvement in detecting new types of anomaly detection [12].

3. FEED FORWARD NEURAL NETWORK (FFNN)

The FFNN allows signals to travel only from input to output. The FFNN tends to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. The FFNN are classified into Single-layer FFNN and Multi-layer FFNN.

The single-layer neural network is the first and the simplest learning machine. The single layer is used to have only two layers such as input layer and output layer. Multi-layer feed forward networks have three layers such as input layer, hidden layer and output layers.

There are two types of phases used in multi layer FFNN, the Forward Phase is used to fix the free parameter in the network and finish with the computation of an error signal.

$$e_i = d_i - y_i \quad (1)$$

where, d_i is the desired response and y_i is the actual output produced by the network in response to the input. In the Backward Phase, the error signal e_i is propagated through the network. During this phase adjustments are applied to the free parameters of the network so as to minimize the error e_i in a statistical sense [13].

4. GENERALIZED REGRESSION NEURAL NETWORK (GRNN)

The General Regression Neural Networks perform regression where the target variable is continuous. If you select a GRNN network, DTREG will automatically select the correct type of network based on the type of target variable. DTREG also provides Multilayer Perceptron Neural Networks and Cascade Correlation Neural Networks.

GRNN networks have advantages and disadvantages compared to Multilayer Perceptron networks:

- It is usually faster to train a GRNN network than a multilayer perceptron network.
- GRNN networks often are more accurate than multilayer perceptron networks.
- GRNN networks are relatively insensitive to outliers.
- GRNN networks are slower than multilayer perceptron networks at classifying new cases.
- GRNN networks require more memory space to store the model.

4.1 GRNN NETWORKS HAVE FOUR LAYERS

Input layer - There is one neuron in the input layer for each predictor variable. In the case of categorical variables, N-1 neurons are used where N is the number of categories. The input neurons then feed the values to each of the neurons in the hidden layer.

Hidden layer – There is one neuron for each case in the training data set. The neuron stores the values of the predictor variables for the case along with the target value. The resulting value is passed to the neurons in the pattern layer.

Pattern layer / Summation layer – There are only two neurons in the pattern layer. One neuron is the denominator summation unit the other is the numerator summation unit. The denominator summation unit adds up the weight values coming from each of the hidden neurons. The numerator summation unit adds up the weight values multiplied by the actual target value for each hidden neuron.

Decision layer – The decision layer divides the value accumulated in the numerator summation unit by the value in the denominator summation unit and uses the result as the predicted target value.

5. PROBABILISTIC NEURAL NETWORK (PNN)

The PNN is a direct continuation of the work on Bayes classifiers. More precisely, the PNN is interpreted as a function which approximates the probability density of the distribution. The PNN consists of nodes allocated in three layers after the

input layers such as pattern layer, summation layer and output layer[16] [17] [18].

A. Pattern Layer:

It is one pattern node for each training phase. Each pattern node forms a product of the pattern vector x for classification and weight vector Wi, Zi = x.Wi (nonlinear operation), the nonlinear operation exp[(Zi-1)/ σ2] is used. Both x and Wi are normalized to unit length, it is equivalent to exp[-(Wi-x)T(Wi-x)/2σ2].

B. Summation Layer:

Each summation node receives the outputs from pattern nodes associated with a given class. It sums the inputs from the pattern node that correspond to the training pattern is selected, ∑i exp[-(Wi-x)T(Wi-x)/2σ2].

C. Output Layer:

The output nodes are two input neurons and units product binary outputs by using the classification criterion: ∑i exp[-(Wi-x)T(Wi-x)/2σ2] > ∑j exp[-(Wj-x)T(Wj-x)/2σ2].

The only factor that needs to be selected for training is the smoothing factor that is the deviation of the Gaussian functions:

- Too small deviations cause a very spiky approximation which cannot be generalized as well.
- Too large deviations smooth out details.
- An appropriate deviation is chosen by experiment.

6. RADIAL BASIS NEURAL NETWORK (RBNN)

A Radial Basis Neural Network (RBNN) has an input layer, a hidden layer and an output layer. The neurons in the hidden layer contain Gaussian transfer functions whose outputs are inversely proportional to the distance from the center of the neuron. The RBNN is viewed as a curve-fitting problem in high-dimensional space. RBF networks have three layers; Input layer, Hidden layer and Summation layer.

The RBF is applied to the distance to compute the weight (influence) for each neuron.

$$\text{Weight} = \text{RBF (distance)} \quad (2)$$

The following parameters are determined by the training process:

- The number of neurons in the hidden layer.
- The coordinates of the center of each hidden-layer RBF function.
- The radius (spread) of each RBF function in each dimension.
- The weights applied to the RBF function outputs as they are passed to the summation layer.

The RBF methods have been used to train the networks. There are two types of approaches used, they are K-means clustering used to find cluster centers which are then used as the centers for the RBF functions and a random subset of the training points as the centers.

7. LAYERED APPROACH FOR INTRUSION DETECTION

The Layer-based Intrusion Detection System (LIDS) is described, this approach is used in the Airport Security model, where a number of security checks are performed one after the other in a sequence basis. Similar to this model, the LIDS represents a sequential Layered Approach and is based on ensuring availability, confidentiality, and integrity of data and (or) services over a network. The goal of using a layered model is to reduce computation and the overall time required to detect anomalous events. Every layer in the LIDS framework is trained separately and then deployed sequentially. We define four layers that correspond to the four attack groups mentioned in the data set. They are Probe layer, DoS layer, R2L layer, and U2R layer. Each layer is then separately trained with a small set of relevant features.

The Fig.1 represents the Layered approach for Intrusion Detection and Layered data as Layer1 for DoS, Layer2 for Probe, Layer3 for R2L and Layer4 for U2R.

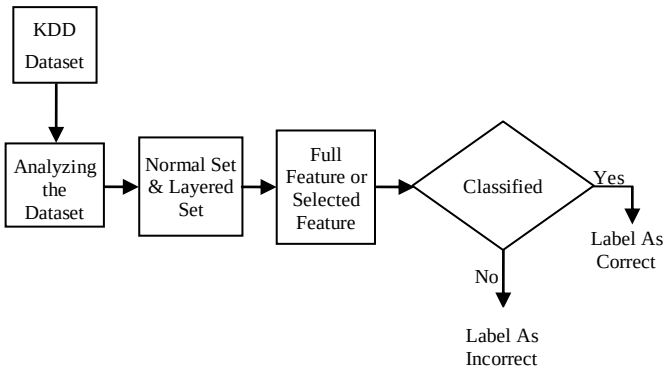

Fig.1. Representation of a Layered Approach

Our second goal is to improve the speed of operation of the system. Hence, we implement the LIDS and select a small set of features for every layer rather than using all the 41 features. This results in significant performance improvement during both the training and the testing of the system. The performance of our proposed system has higher attack detection accuracy.

8. KDD CUP 1999 DATASET DESCRIPTION

The KDD Cup 1999 dataset has been used for the evaluation of intrusion detection methods. The KDD Cup 1999 training dataset consists of approximately 4,900,000 single connection vectors each of which contains 41 features and is labeled as either normal or an attack, with exactly one specific attack type [19].

In KDD Cup 1999 dataset has the different types of attacks: back, buffer_overflow, ftp_write, guess_passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, normal, perl, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster. The datasets contain a total number of 24 training attack types, with an additional 14 types in the test data only. These attacks can be divided into 4 groups [14].

The Table.1 shows the list of attacks in category wise:

Table.1. List of attacks - category wise

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

Denial of Service (DoS) attacks: deny legitimate requests to a system, e.g. flood, User-to-Root (U2R) attacks: unauthorized access to local super user(root) privileges, e.g. various buffer overflow attacks, Remote-to-Local (R2L) attacks: unauthorized access from a remote machine, e.g. guessing password, and Probing: surveillance and other probing, e.g. port scanning [9].

The sets are named as A, B, C, D, and E respectively. The set 'A' acquires data from DoS class. The set 'B' acquires data from U2R class. The set 'C' acquires data from R2L class. The set 'D' acquires data from Probe Class. The set 'E' acquires data from Normal class. The following sets of data can be used for training and testing the data from KDD Cup 1999 dataset.

Table.2. Training and Testing Data Set

	Training Set	Testing Set
DoS	300	300
U2R	20	19
R2L	300	300
Probe	300	300
Normal	300	300
Total	1220	1219

The 41 featured dataset and reduced featured dataset for each class is used to detect the attacks in KDD Cup 1999 dataset. The 41 features are listed in the website [19].

For converting symbols into numerical form, an integer code is assigned to each symbol. For instance, in the case of protocol_type feature, 0 is assigned to tcp, 1 to udp, and 2 to the icmp symbol and so on. Attack names are first mapped to one of the five classes, 'A' for DoS, 'B' for U2R, 'C' for R2L, 'D' for Probe and 'E' for Normal. Two features spanned over a very large integer range, namely src_bytes [0, 1.3 billion] and dst_bytes [0, 1.3 billion]. Logarithmic scaling (with base 10) is applied to these features to reduce the range to [0.0, 9.14]. All other features are Boolean, in the range [0.0, 1.0]. Hence scaling is not necessary for these attributes.

300 signals from DoS, R2L, Probe and Normal class each and 20 signals from U2R class are selected for training the network. Four different neural networks are used for training the KDD Cup 1999 data. The networks are usually trained to perform tasks such as pattern recognition and decision-making. The Table.2 represents the training set.

300 signals from DoS, R2L, Probe and Normal class each and 19 signals from U2R class are selected for testing the network. Four different neural networks are used for testing the KDD Cup 1999 data. By testing the KDD Cup 1999 data, the accuracy of the each neural networks are measured. The Table.2 represents the testing set [13].

9. EXPERIMENTAL RESULTS

The Layered Neural Network techniques are used to detect the intrusions based on the KDD Cup 1999 dataset. These dataset contain 41 features in various types of attacks. By reducing 41 features into 5 features for Probe Layer, 9 features for DoS Layer, 14 features for R2L Layer and 8 features for U2R Layer. These Dataset can be applied using MATLAB software [20] and comparing these four Neural Network classifiers with best, average and worst results.

Normally, neural network or data mining techniques are used to address the intrusion detection system because these soft computing mechanism which perform accurate and faster. The parameters are very important to measure the intrusions.

For our results, measure the Precision, Recall, F-Value accuracy and False Alarm Rate (FAR) with the given data set, it is easy to achieve high accuracy by carefully selecting the sample size [1]. The comparison of intrusion detection system uses all 41 variables which give reasonable precision value, recall value, f-value, and efficiency and minimize the false alarm rate. The neural networks which performs better than the other algorithms.

From Table.2, we note that the number of sample instances is very low. Hence, if we use accuracy as a measure for testing the performance of the system, the system can be biased and can attain an accuracy of 100 percent for DoS attacks. However, Precision, Recall, and F-Value are not dependent on the size of the training and the test samples.

There are different metrics are used to measure the performance namely Precision, Recall, F-value, Efficiency and False Alarm Rate (FAR) by using confusion matrix.

	Classified as Normal	Classified as Attack
Normal	TP	FP
Attack	FN	TN

where,

TP – denotes the number of connections classified as Normal while they actually were Normal.

TN – denotes the number of connections classified as Attack while they actually were Attack.

FP – denotes the number of connections classified as Attack while they actually were Normal.

FN – denotes the number of connections classified as Normal while they actually were Attack.

They are defined as follows:

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FP} \quad (4)$$

$$F-Value = \frac{\left(1+\beta^2\right)* Recall * Precision}{\beta^2 * \left(Recall * Precision\right)} \quad (5)$$

$$Efficiency = \frac{Total\_Detected\_Attack}{Total\_Attacks} * 100 \quad (6)$$

False Alarm Rate (FAR): It is the ratio between total numbers of misclassified instances to the total number of normal instances.

$$FAR = \frac{Total\ misclassified\ instances}{Total\ normal\ instances} * 100 \quad (7)$$

where, TP, FP, and FN are the number of True Positives, False Positives, and False Negatives, respectively, and β corresponds to the relative importance of precision versus recall and is usually set to 1.

9.1 41 FEATURES AND REDUCED DATASET FOR DOS

The following table contains the four types of classifiers and compared with 41 features dataset and reduced features dataset which is represented in Appendix A (A.1) is used and the efficiency is measured. The Table.3 shows the classification of 41 featured and reduced dataset for DoS.

Here the classification of KDD Cup'99 data set has been performed using 41 features dataset and 9 features dataset for DoS. The accuracy of 41 features datasets: accuracy of FFNN, GRNN and PNN are 100%; and accuracy of RBNN is 99%. The accuracy of reduced features datasets: accuracy of FFNN is 99.17%; accuracy of GRNN and PNN are 100%; and accuracy of RBNN is 99.3%. Also measures the False Alarm Rate (FAR) is 0.00% for 41 features and reduced features.

Table.3. Results for 41 Features and Reduced Dataset for DoS

Functions		Features	Precision	Recall	F-Value	Efficiency%	FAR %
Feed Forward Neural Network (FFNN)	Best	41	100	100	100	100	0.00
	Average		99.34	100	99.67	99.67	0.33
	Worst		99.34	100	99.67	99.67	0.33
	Best	9	98.68	99.67	99.17	99.17	0.83
	Average		98.04	100	99.01	99	1.00
	Worst		96.77	100	98.36	98.33	1.67
Generalized Regression Neural Network (GRNN)	Best	41	100	100	100	100	0.00
	Average		99.67	100	99.83	99.83	0.17
	Worst		99.67	99.33	99.5	99.5	0.5
	Best	9	100	100	100	100	0.00
	Average		100	99.33	99.67	99.67	0.33
	Worst		100	99	99.5	99.5	0.5
Probabilistic Neural Network (PNN)	Best	41	100	100	100	100	0.00
	Average		100	97.33	98.65	98.67	1.33
	Worst		100	96.67	98.31	98.33	1.67
	Best	9	100	100	100	100	0.00
	Average		100	99.33	99.67	99.67	0.33
	Worst		100	99	99.5	99.5	0.5
Radial Basis Neural Network (RBNN)	Best	41	98.04	100	99.01	99	1.0
	Average		100	97.33	98.65	98.67	1.33
	Worst		100	69.67	98.31	98.33	1.67
	Best	9	99.66	99	99.33	99.33	0.67
	Average		100	98.33	99.16	99.17	0.83
	Worst		100	98	98.99	99	1.00

9.2 41 FEATURES AND REDUCED DATASET FOR PROBE

The following table contains the four types of classifiers and compared with 41 features dataset and reduced features dataset which is represented in Appendix A (A.2) is used and the efficiency is measured. The Table.4 shows the classification of 41 featured and reduced dataset for Probe.

Here the classification of KDD Cup '99 data set has been performed using 41 features dataset and 5 features dataset for Probe. The accuracy of 41 features datasets: accuracy of FFNN is 94.33%; accuracy of GRNN is 98.33%; accuracy of PNN is 99.17%; and accuracy of RBNN is 82.17%. The accuracy of reduced features datasets: accuracy of FFNN is 50.83%; accuracy of GRNN is 98.5%; accuracy of PNN is 99.5%; and accuracy of RBNN is 91.67%. Also a measure the False Alarm Rate (FAR) for 41 features is 0.83% and reduced features is 0.5%.

Table.4. Results for 41 Features and Reduced Dataset for Probe

Functions		Features	Precision	Recall	F-Value	Efficiency%	FAR %
Feed Forward Neural Network (FFNN)	Best	41	98.19	90.33	94.1	94.33	5.67
	Average		98.49	86.67	92.2	92.67	7.33
	Worst		98.49	86.67	92.2	92.67	7.33
	Best	5	85.71	2	3.9	50.83	49.17
	Average		85.71	2	3.9	50.83	49.17
	Worst		75	2	3.9	50.67	49.33
Generalized Regression Neural Network (GRNN)	Best	41	96.78	100	98.36	98.33	1.67
	Average		96.46	100	98.2	98.17	1.83
	Worst		95.85	100	97.88	97.83	2.17
	Best	5	97.09	100	98.52	98.5	1.50
	Average		95.85	100	97.88	97.83	2.17
	Worst		95.54	100	97.72	97.67	2.33
Probabilistic Neural Network (PNN)	Best	41	98.36	100	99.17	99.17	0.83
	Average		98.04	100	99.01	99	1.00
	Worst		97.4	100	98.68	98.67	1.33
	Best	5	99.01	100	99.5	99.5	0.50
	Average		98.36	100	99.17	99.17	0.83
	Worst		97.09	100	98.52	98.5	1.50
Radial Basis Neural Network (RBNN)	Best	41	84.84	78.33	81.46	82.17	17.83
	Average		86.92	75.33	80.71	82	18.00
	Worst		84.19	76.33	80.07	81	19.00
	Best	5	85.71	100	92.31	91.67	8.33
	Average		85.47	100	92.17	91.5	8.50
	Worst		84.74	100	91.74	91	9.00

9.3 41 FEATURES AND REDUCED DATASET FOR R2L

The following table contains the four types of classifiers and compared with 41 features dataset and reduced features dataset which is represented in Appendix A (A.3) is used and the efficiency is measured. The Table.5 shows the classification of 41 featured and reduced dataset for R2L.

Here the classification of KDD Cup '99 data set has been performed using 41 features dataset and 14 features dataset for R2L. The accuracy of 41 features datasets: accuracy of FFNN is 98.83%; accuracy of GRNN is 57.17%; accuracy of PNN is 92.5%; and accuracy of RBNN is 97%. The accuracy of reduced features datasets: accuracy of FFNN is 99.5%; accuracy of GRNN is 97.5%; accuracy of PNN is 92.5%; and accuracy of RBNN is 99.33%. Also a measure the False Alarm Rate (FAR) for 41 features is 1.17% and reduced features is 0.5%.

Table.5. Results for 41 Features and Reduced Dataset for R2L

Functions		Features	Precision	Recall	F-Value	Efficiency%	FAR %
Feed Forward Neural Network (FFNN)	Best	41	98.35	99.33	98.84	98.83	1.17
	Average		97.71	99.33	98.51	98.5	1.50
	Worst		97.36	98.33	97.84	97.83	2.17
	Best	14	99.01	100	99.5	99.5	0.50
	Average		99.01	100	99.5	99.5	0.50
	Worst		98.68	100	99.34	99.33	0.67
Generalized Regression Neural Network (GRNN)	Best	41	76.54	20.67	32.55	57.17	42.83
	Average		80	18.67	30.27	57	43.00
	Worst		74.7	20.67	32.38	56.83	43.17
	Best	14	95.24	100	97.56	97.5	2.50
	Average		94.94	100	97.4	97.33	2.67
	Worst		92.88	100	96.31	96.17	3.83
Probabilistic Neural Network (PNN)	Best	41	86.96	100	93.02	92.5	7.50
	Average		83.33	100	90.91	90	10.00
	Worst		81.97	100	90.09	89	11.00
	Best	14	86.96	100	93.02	92.5	7.50
	Average		86.46	100	92.74	92.17	7.83
	Worst		83.33	100	90.9	90	10.00
Radial Basis Neural Network (RBNN)	Best	41	94.34	100	97.09	97	3.00
	Average		86.96	100	93.02	92.5	7.50
	Worst		78.74	100	88.11	86.5	13.50
	Best	14	99.33	99.33	99.33	99.33	0.67
	Average		99.33	99	99.17	99.17	0.83
	Worst		95.56	100	97.56	97.5	2.50

9.4 41 FEATURES AND REDUCED DATASET FOR U2R

The following table contains the four types of classifiers and compared with 41 features dataset and reduced features dataset which is represented in Appendix A (A.4) is used and the efficiency is measured. The Table.6 shows the classification of 41 featured and reduced dataset for U2R.

Here the classification of KDD Cup '99 data set has been performed using 41 features dataset and 8 features dataset for U2R. The accuracy of 41 features datasets: accuracy of FFNN is 96.24%; accuracy of GRNN is 94.67%; accuracy of PNN is 94.98%; and accuracy of RBNN is 94.04%. The accuracy of reduced features datasets: accuracy of FFNN is 95.3%; accuracy of GRNN is 95.61%; and accuracy of PNN and RBNN are 96.24%. Also a measure the False Alarm Rate (FAR) for 41 features is 3.76% and reduced features is 4.39%.

9.5 COMPARISON OF RESULTS USING VARIOUS ALGORITHMS

Experimental results has been analyzed and compared in Table.7, from experimental results to conclude that the FFNN may be very effective in detecting the DoS, the R2L, and the U2L attacks and also reduce the False Alarm Rate (FAR) compared with other algorithms [15]. The proposed algorithm uses for KDD 99 data set for 41 features.

Normally, neural network or data mining techniques are used to address the intrusion detection system because these soft computing mechanisms perform accurate and faster. In order to critically analyze and compare the performance of various intrusion detection system using parameters such as Precision, Recall, F-value, Efficiency and False alarm rate, extensive experimentation is done and presented in this paper. The parameters are very important to measure the intrusions.

Table.6. Results for 41 Features and Reduced Dataset for U2R

Functions		Features	Precision	Recall	F-Value	Efficiency%	FAR %
Feed Forward Neural Network (FFNN)	Best	41	73.33	57.9	64.71	96.24	3.76
	Average		72.73	42.11	53.33	95.61	4.39
	Worst		62.5	52.63	57.14	95.3	4.70
	Best	8	100	21.05	34.78	95.3	4.70
	Average		100	21.05	34.78	95.3	4.70
	Worst		71.43	26.32	38.46	94.98	5.02
Generalized Regression Neural Network (GRNN)	Best	41	75	15.8	26.09	94.67	5.33
	Average		60	15.79	25	94.36	5.64
	Worst		50	15.8	24	94.04	5.96
	Best	8	100	26.32	41.67	95.61	4.39
	Average		100	10.53	19.05	94.67	5.33
	Worst		100	5.26	10	94.36	5.64
Probabilistic Neural Network (PNN)	Best	41	100	15.79	27.27	94.98	5.02
	Average		100	10.53	19.05	94.67	5.33
	Worst		75	15.79	26.09	94.67	5.33
	Best	8	100	36.84	53.85	96.24	3.76
	Average		100	26.32	41.67	95.61	4.39
	Worst		100	15.79	27.27	94.98	5.02
Radial Basis Neural Network (RBNN)	Best	41	1	1	1	94.04	5.96
	Average		1	1	1	93.73	6.27
	Worst		29.41	26.32	27.78	91.85	8.15
	Best	8	76.92	52.63	62.5	96.24	3.76
	Average		100	31.58	48	95.92	4.08
	Worst		100	26.32	41.67	95.61	4.39

With the change in the usage of parameter, the results vary, so the user has to select the parameter according to their attacks.

When compared with various algorithms, the different algorithm gives different values to measure the kinds of attacks. The comparison of intrusion detection system uses all 41 variables which give reasonable precision value, recall value, f-value, and efficiency and minimize the false alarm rate. The neural networks perform better than the other algorithms.

Table.7. Performance comparison of various Algorithms

Classifier Algorithms	Performance Measures in %				
	Precision %	Recall %	F-value %	Efficiency %	FAR %
Feed Forward Neural Network	92.47	86.89	89.41	97.35	2.65
Generalized Regression Neural Network	87.08	59.12	64.25	87.54	12.46
Probabilistic Neural Network	96.33	78.95	79.87	96.66	3.34
Radial Basis Neural Network	69.56	69.83	69.64	93.05	6.95
K-Means	90.3	-	84.2	89.4	5.7
ID3	93.1	-	91.7	93.0	4.3
Naïve Bayes	92.5	-	91.5	93.2	4.2
SVM	90.7	-	92.3	95.5	2.7
K-Means + C4.5	95.6	-	94.0	95.8	0.1

10. CONCLUSION

A Layered Neural Network approach for detecting network intrusions using four classifiers are proposed in this paper. This study proves that the FFNN, GRNN and PNN provide better accuracy over other approaches for DoS attack. The PNN provides better accuracy over other approaches for Probe attack. The FFNN provides better accuracy over other approaches for R2L attacks and U2R attacks. These approaches are applied to the KDD Cup 1999 dataset using MATLAB software. Comparing these four classifiers FFNN gives better efficiency than GRNN, PNN and RBNN for DoS attack, R2L attack and U2R attack. The overall efficiency of Feed Forward Neural Network (FFNN) measures 97.35% when compared with various algorithms. Hence, it is proposed to consider FFNN techniques to improve the efficiency and reduce the false alarm rate.

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

A.4 Features Selected for U2R Layer:

Feature Number	Feature Name
10	hot
13	num_compromised
14	root_shell
16	num_root
17	num_file_creations
18	num_shells
19	num_access_files
21	is_host_login

REFERENCES

[1] K.K. Gupta, B. Nath and R. Kotagiri, "Layered Approach Using Conditional Random Fields for Intrusion Detection", *IEEE Transactions on Dependable and Secure Computing*, Vol.7, No.1, pp. 35-49, 2010.

[2] Nong Ye, Syed Masum Emran, Qiang Chen and Sean Vilbert, "Multivariate Statistical Analysis of audit Trails for Host-Based Intrusion Detection", *IEEE Transactions on Computers*, Vol. 51, No. 7, pp. 810-820, 2002.

[3] Jiankun Hu, Xinghuo Yu, D. Qiu and Hsiao-Hwa Chen, "A simple and efficient hidden Markov model scheme for host- based anomaly intrusion detection", *IEEE Network*, Vol. 23, No. 1, pp. 42-47, 2009.

[4] Yi Xie and Shun-Zheng Yu, "A Large Scale Hidden Semi-Markov model for Anomaly Detection on User Browsing Behaviors", *IEEE/ACM Transactions on Networking*, Vol. 17, No. 1, pp. 1-14, 2009.

[5] Mansour Sheikhan, Zahra Jadidi and Ali Farrokhi, "Intrusion detection using reduced-size RNN based on feature grouping", *Neural Computing and Applications*, Vol. 21, No. 6, pp. 1185-1190, 2010.

[6] Wei Li, "Using Genetic Algorithm for network intrusion detection", *Proceedings of the United States Department of Energy Cyber Security Group 2004 Training Conference*, pp. 24-27, 2004.

[7] Jiang Hua and Ruan Junhu, "The Application of Genetic Neural Network in Network Intrusion Detection", *Journal of Computers*, Vol. 4, No. 12, pp. 1223-1230, 2009.

[8] Neveen I. Ghali, "Feature Selection for Effective Anomaly-Based Intrusion Detection", *International Journal of Computer Science and Network Security*, Vol. 9, No. 3, pp. 285-289, 2009.

[9] V. Venkatachalam and S. Selvan, "Intrusion detection using an improved competitive learning lamstar neural network", *International Journal of Computer Science and Network Security*, Vol. 7, No. 2, pp. 255-259, 2007.

[10] Suseela T. Sarasamma, A. Qiuming, Q.A. Zhu and Julie Huff, "Hierarchical Kohonenen Net for Anomaly Detection in Network Security", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 35, No. 2, pp. 302-312, 2005.

[11] Ning Chen, Xiao-Su Chen, Bing Xiong and Hong-Wei Lu, "An Anomaly Detection and Analysis Method for Network Traffic Based on Correlation Coefficient Matrix", *IEEE International Conference on Scalable Computing and Communication; Eighth IEEE International Conference on Embedded Computing*, pp. 238-244, 2009.

[12] E. Anbalagan, C. Puttamadappa, E. Mohan, B. Jayaraman and Srinivasarao Madane, "Datamining and Intrusion Detection Using Back-Propagation Algorithm for Intrusion Detection", *International Journal of Soft Computing*, Vol. 3, No. 4, pp. 264-270, 2008.

[13] S. Devaraju and S. Ramakrishnan, "Performance Comparison of Intrusion Detection System using Various Techniques – A Review", *ICTACT Journal on Communication Technology*, Vol. 4, No. 3, pp. 802-812, 2013.

[14] S. Devaraju and S. Ramakrishnan, "Performance Analysis of Intrusion Detection System Using Various Neural Network Classifiers", *International Conference on International Conference on Recent Trends in Information Technology*, pp. 1033-1038, 2011.

[15] Amuthan Prabakar Muniyandi, R. Rajeswari and R. Rajaram, "Network Anomaly Detection by Cascading K-Means Clustering and C4.5 Decision Tree algorithm", *International Conference on Communication Technology and System Design, Elsevier Procedia Engineering*, Vol. 30, pp. 174-182, 2012.

[16] S. Ramakrishnan and Ibrahiem M.M. El Emary, "Classification brain MR images through a fuzzy multiwavelets based GMM and probabilistic neural networks", *Telecommunication Systems*, Vol. 46, No. 3, pp. 245-252, 2011.

[17] Ibrahiem M.M. El Emary and S. Ramakrishnan, "On the Application of Various Probabilistic Neural Networks in Solving Different Pattern Classification Problems", *World Applied Sciences*, Vol. 4, No. 6, pp. 772-780, 2008.

[18] P.V. Nageswara Rao, T. Uma Devi, D.S.V.G.K. Kaladhar, G.R. Sridhar and Allam Appa Rao, "A Probabilistic Neural Network Approach for Protein Superfamily Classification", *Journal of Theoretical and Applied Information Technology*, Vol. 6, No. 1, pp. 101-105, 2009.

[19] The UCI KDD Archive, "KDD Cup 1999 Data", Information and Computer Science, 1999.

[20] MATLAB (MATrix Laboratory) tutorials, http://terpconnect.umd.edu/~nsw/ench250/matlab.htm