A Comparative Study of Different Artificial Neural Networks Based Intrusion Detection Systems

Afrah Nazir

Computer Engineering Section, Women's Polytechnic, A. M. U. Aligarh, 202002, India

Abstract- Information is an important asset of an organization. Large amount of information need to be stored and processed in network based computers. The confidentiality, integrity and availability of the system resources have raised the vulnerability of these systems to security threats, attacks and intrusions. One idea is to use a neural network algorithm for detecting intrusions. The neural network algorithms are popular for their ability to 'learn' the patterns in a given environment and thus can be trained to detect intrusions by recognizing patterns of an intrusion. In this work we perform a comparative study of Multilayer Feed Forward, Elman Back Propagation, Cascaded Forward Back Propagation and Self Organizing Feature Map neural networks based intrusion detection systems. In this study we work on the well structured KDD CUP 99 dataset.

Index Terms- Multilayer Feed Forward (MLFF), Self Organizing Feature Map (SOFM), Elman Back Propagation (ELBP), Cascaded Forward Back Propagation (CFBP) Intrusion Detection Systems (IDSs).

I. Introduction

In a network based computers large amount of information need to be stored and processed and thereby raising their vulnerabilities to attacks. An intrusion is defined as any set of actions that compromise the integrity, confidentiality or availability of a resource in a system. Data integrity is the assurance that data received are exactly as sent by an authorized entity (i.e. contain no modification, insertion, deletion, or replay). Data confidentiality is the protection of data from unauthorized disclosure. Availability of resource or access control is the prevention of unauthorized use of resource.

An intrusion detection system inspects all inbound and outbound network activities and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. The learning ability of neural networks can be used in connection with an intrusion detection system, where the neural network algorithm can be trained to detect intrusions by recognizing patterns of an intrusion.

This work compares and evaluates the performance of Multilayer Feed Forward, Elman Back Propagation and Cascaded Forward Back Propagation neural networks approaches to intrusion detection based on classification rate, false positive rate and false negative rate for each of the four classes of attacks present in KDD CUP 99 Dataset.

This paper is organized as follows. In next section, we discuss related work in the area of intrusion detection using Neural Networks. In the next four sections, we discuss the details of different neural networks used for intrusion detection in this study, namely, Self Organizing Feature Map, Multilayer Feed Forward Neural Network, Elman Back Propagation, Cascaded Forward Back Propagation and Neural Network. In section 7, we discuss the training and test datasets used in this study. In section 8, we discuss the various performance evaluation criteria. In section 9, we show our system's results and finally conclude in last section.

II. Related Works

The intrusion detection systems operate by making results in the sense of predictions based on known as well as unknown patterns. With the use of neural network models it is possible to comply with this process, since these models offer the option to train a custom network and use it as some sort of a trainer for new incoming network connection and thereby detect abnormal behaviors.

A Multilayer Perceptron (MLP) was used in [1] for anomaly detection. The proposed model is a single hidden layer neural network. The performance of this model tested on the DARPA 1998 data set was a correct detection rate of 77%with 2.2%false alarms.

A Self Organizing Map was selected in [2] for intrusion detection. In that work, the self-organizing map was designed to learn the characteristics of normal activities. The variations from normal activities provided an indication of a virus.

In [9], SOM was used to map the network connections onto 2-dimensional surfaces, which were displayed to the network administrator. The intrusions were easily detected in this view. However, the approach needs a visual interpretation by the network administrator.

In this work, we develop and perform a comparative study of different neural networks based IDSs.

III. Kohonen's Self Organizing Feature Maps (SOFM)

The model was first described by the professor Teuvo Kohonen and is thus sometimes referred to as a Kohonen Map. The SOFM is often used in the fields of data compression and pattern recognition. The structure of the SOFM is a single feed forward network, where each source node of the input layer is connected to all output neurons. The number of the input dimensions is usually higher than the output dimension. The neurons of the Kohonen layer in the SOFM are organized into a grid. The training utilizes competitive learning, meaning that neuron with weight vector that is most similar to the input vector

is adjusted towards the input vector. This neuron is said to be the 'winning neuron' or the Best Matching Unit (BMU).

The weights of the neurons close to the winning neuron are also adjusted but the magnitude of the change depends on the physical distance from the winning neuron and it is also decreased with the time. The algorithm tries to find clusters such that two neighboring clusters in the grid have codebook vectors close to each other in the input space. Another way to look at this is that related data in the input data set are grouped in clusters in the grid.

The algorithm proposed by Kohonen follows two basic equations: matching and finding the winner neuron determined by the minimum Euclidean distance to the input. The topological ordering property has induced in adopting Self- Organizing Feature Map for intrusion detection.

IV. Multilayer Feed Forward Neural Network (MLFF)

Artificial Neural Networks can be viewed as parallel and distributed processing systems which consists of a huge number of simple and massively connected processors. The MLP architecture is the most popular paradigm of artificial neural networks in use today. The neural network architecture in this class share common feature that are as follows: -

All neurons in a layer are connected to all neurons in adjacent layers through unidirectional branches. That is, the branches and links can only broad information in one direction, that is, the "forward direction". The branches have associated weights that can be adjusted according to a defined learning rule.

Feed forward neural network training is usually carried out using the called BPA. Hence it is also called by the name Back Propagation Network (BPN). Training the network with BPA results in a non-linear mapping between the input and output variables. Thus, given input/output pairs, the network can have its weights adjusted by the BPA to capture the non-linear relationship. After training, the networks with fixed weights provide the output for the given input.

V. Elman Back Propagation Neural Network (ELBP)

A strict feed forward architecture does not maintain a short-term memory. Any memory effects are due to the way past inputs are represented to the network (as for the tapped delay line). A simple recurrent network, Elman network, has activation feedback which embodies short-term memory.

A state layer is updated not only with the external input of the network but also with activation from the previous forward propagation. The feedback is modified by a set of weights as to enable automatic adaptation through learning (e.g. back propagation).

In the Elman neural network, if it is a fully recurrent network, every neuron receives inputs from every other neuron in the network. These networks are not arranged in layers. Usually only a subset of the neurons receive external inputs in addition to the inputs from all the other neurons, and another disjoint subset of neurons report their output externally as well as sending it to all the neurons. These distinctive inputs and outputs perform the function of the input and output layers of a simple recurrent network, and also join all the other neurons in the recurrent processing.

VI. Cascaded Forward Back Propagation Neural Network (CFBP)

A cascade correlation net consists of input units, hidden units, and output units. Input units are connected directly to output units with adjustable weighted connections. Connections from inputs to a hidden unit are trained when the hidden unit is added to the net and are then frozen. Connections from the hidden units to the output units are adjustable consequently.

Cascade correlation network starts with a minimal topology, consisting only of the required input and output units. This net is trained until no further improvement is obtained. The error for each output until is then computed. Next, one hidden unit is added to the net in a two-step process. During the first step, a candidate unit is connected to each of the input units, but is not connected to the output units. The weights on the connections from the input units to the candidate unit are adjusted to maximize the correlation between the candidate's output and the residual error at the output units.

The residual error is the difference between the target and the computed output, multiplied by the derivative of the output unit's activation function, i.e., the quantity that would be propagated back from the output units in the back propagation algorithm. When this training is completed, the weights are frozen and the candidate unit becomes a hidden unit in the net.

The second step in which the new unit is added to the net now begins. A second hidden unit is added using the same process. The process of adding a new unit, training its weights from the input units and the previously added hidden units, and then freezing the weights, followed by training all connections to the output units, is continued until the error reaches an acceptable level or the maximum number of epochs (or hidden units) is reached.

VII. Training and Testing Datasets

The dataset used in this study for intrusion detection is the KDD Cup 99 Dataset [9]. The KDD 99 intrusion detection datasets are based on the 1998 DARPA [7] initiative, which provides designers of intrusion detection systems (IDS) with a benchmark on which to evaluate different methodologies.

7.1 Categories of Attacks in KDD 99 Dataset

Attacks in KDD 99 Dataset fall in the following four major categories:

Denial of Service (DoS) attacks: - DoS is a class of attacks where an attacker makes some computing or memory resource too busy or too full to handle legitimate requests, thus denying legitimate users access to a machine There are different ways to launch DoS attacks: by abusing the computer's legitimate features, by targeting the implementations bugs; or by exploiting the system's misconfigurations. Smurf, Teardrop, Neptune, pod are the common DoS attacks.

Probe: - Probe is a class of attacks where an attacker scans a network to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use this information to look for exploits. Portsweep and Satan are the common Probing attacks.

User to root attacks (U2R): - User to root exploits are a class of attacks where an attacker starts out with an access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system. Rootkit and perl is the User to Root attacks.

Remote to user attacks (R2L): - A remote to user is a class of attacks where an attacker sends packets to a machine over a network, then exploits machine's vulnerability to illegally gain local access as a user. Multihop, Spy, is the common Remote to User attacks.

7.2 Training Dataset

The "10 % KDD" dataset is used for the training of different intrusion detection systems. It includes 22 types of attacks connections. This dataset contains 22 attack types and is a more concise version of the "Whole KDD" dataset. It contains more examples of attacks than normal connections and the attack types are not represented equally. Denial of Service attack accounts for the majority of records of the dataset.

7.3 Test Dataset

The "Corrected KDD" dataset is used for testing purpose. The "Corrected KDD" dataset provides a data with different statistical distributions as compared to the data present in either "10% KDD" or "Whole KDD" datasets. Labels for each of the connection records are used to verify our classification predictions made during testing.

VIII. Evaluation Criteria

Following are the different criteria for performance evaluation of the different neural networks based intrusion detection systems.

8.1 Classification Rate (CR)

It denotes true-positives rate or true-negatives rate. An intrusion detection system gets more accurate as it detects more attacks and raises fewer false alarms. Or in other words its classification accuracy or classification rate is high.

$$\frac{\textit{True-positives} = \textit{Total Number of Normal Instances detected \& classified by the system}}{\textit{Total Number of Normal Instances present in the Test Dataset}}$$

$$\frac{\textit{True-negatives} = \textit{Total Number of Attack Instances detected \& classified by the system}}{\textit{Total Number of Attack Instances present in the Test Dataset}}$$

Classification Percent is True-positives x 100 or True-negatives x 100.

8.2 Mean Square Error

Mean Square Error (MSE) is the squared prediction error. Lesser the MSE the better the classification rate of the network, this means less number of false classification.

8.3 Confusion Matrix

Network Intrusion detection is a two-class classification problem i.e. classify a connection record to a normal or an attack connection. Its effectiveness can be defined as the ability to make correct predictions of samples. Table 1 shows the confusion matrix for Normal-Attack Classification problem.

Table 1 Confusion Matrix

Predicted → / **Actual ↓**	**Negative**	**Positive**
Negative	a	b
Positive	c	d

False-Positive Rate = b / (a + b)
False-Negative Rate = c / (c + d)

A False-positive (FPR) occurs when the system classifies an action as anomalous (a possible intrusion) when it is a legitimate action. Although this type of error may not be completely eliminated, a good system should minimize its occurrence.

A False-negative (FNR) occurs when an actual intrusive action has occurred but system allows it to pass as non-intrusive behavior. This implies malicious data is not detected and alerted. It is a more serious error.

8.4 Training Time

Training time is the time required to train the network according to the parameters set for training. It is measured in seconds. In this study training time for each of the 5-Folds of the 5-Fold Cross Validation training approach is calculated for each of the Sub-dataset of the specific connection type (normal, DoS, R2L, U2R).

IX. Experimental Results of Different Neural Network Based Intrusion Detection Systems

In the previous section, we described the various performance evaluation criteria. Now, we list the results of comparison of the four different neural networks intrusion detection systems namely: (i) MLFF, (ii) CFBP, (iii) ELBP and (iv) SOFM based IDS, on the basis of these performance evaluation criteria.

9.1 Results with Different Training Functions

Here in this section, the results are shown for testing new untrained connection records on the "41 41 40 1" neural architecture. This implies that each of the neural network

intrusion detection systems contains 41 nodes in input layer, 41 nodes in first hidden layer, 40 nodes in second hidden layer and 1 node in output layer. The MLFF, CFBP and ELBP are evaluated on this architecture with the training parameters as listed in Table 2.

Table 2 Training Parameters for "41 41 40 1" Neural Network Architecture

Parameters for training	**Value of the parameter**
Number of Nodes in Input Layer	41
Number of Nodes in Output Layer	1
Number of Nodes in First Hidden Layer	41
Number of nodes in second Hidden Layer	40
Training Functions	trainoss OR traingdx OR trainrp
Activation function for HL1, HL2 & OL	tansig, logsig & purelin
Learning Rate and Number of Epochs	0.2 & 1000

1. CFBP Neural Network Intrusion Detection System: The Average Classification Rate for CFBP Neural Network is shown in Table 3. The percentage of normal or attack connections detected by CFBP network using "trainrp" function is the higher of the other two functions. Like for example CR (%) for correctly classified data of class DoS is 90.03 % with "trainoss" function, 92.76 % with "traingdx" function and 94.73 % with "trainrp" function. However the classification rate of R2L with "trainoss" training function is good as compared to classification rate (%) of 81.84 with "traingdx" function, but still the classification percentage for R2L is better with "trainrp" function.

Table 3 Average Classification Rate for CFBP Neural Network

Training function	**Class**	**Classification Rate (%)**
Trainoss	Normal	92.77
	DoS	90.03
	Probe	90.23
	U2R	81.23
	R2L	84.73
Traingdx	Normal	92.83
	DoS	92.76
	Probe	93.08
	U2R	86.47
	R2L	81.84
Trainrp	Normal	93.86
	DoS	94.73
	Probe	94.05
	U2R	90.08
	R2L	92.36

2. MLFF Neural Network Intrusion Detection System: The Average Classification Rate for MLFF Neural Network is given in Table 4. The percentage of normal or attack connections detected by MLFF network using "trainrp" function is the higher of the other two functions. Like for example the classification for attack type "DoS" is 97.78 % using "trainoss" training function, while it is 96.85 using "traingdx" training function and 97.88 % using "trainrp" training function. This shows that for the attack type DoS the performance of "trainoss" training function is better than the "traingdx" training function.

Table 4 Average Classification Rate for MLFF Neural Network

Training function	**Class**	**Classification Rate (%)**
Trainoss	Normal	96.44
	DoS	97.78
	Probe	97.38
	U2R	90.86
	R2L	91.01
Traingdx	Normal	96.75
	DoS	96.85
	Probe	96.91

	U2R	89.95
	R2L	92.06
Trainrp	Normal	97.75
	DoS	97.88
	Probe	97.92
	U2R	90.88
	R2L	95.46

3. **ELBP Neural Network Intrusion Detection System:** The Average Classification Rate for ELBP Neural Network is given in Table 5. Classification rate with "trainrp" is better of other two training functions.

Table 5 Average Classification Rate for ELBP Neural Network

Training function	Class	Classification Rate (%)
Trainoss	Normal	97.75
	DoS	97.95
	Probe	97.95
	U2R	92.89
	R2L	96.76
Traingdx	Normal	98.57
	DoS	97.98
	Probe	98.81
	U2R	94.56
	R2L	97.66
Trainrp	Normal	98.59
	DoS	99.95
	Probe	98.91
	U2R	94.95
	R2L	97.86

4. Performance of Different Neural Networks based IDSs using "trainoss" function: Figure 1 shows, the performance of ELBP network is better than other two networks using "trainoss" function. For the classes normal, DoS and Probe the performances of ELBP and MLFF neural networks in terms of classification rate are very close. CR (%) of attack type Probe is 97.38 % using MLFF neural network, while for ELBP it is 97.95 %.

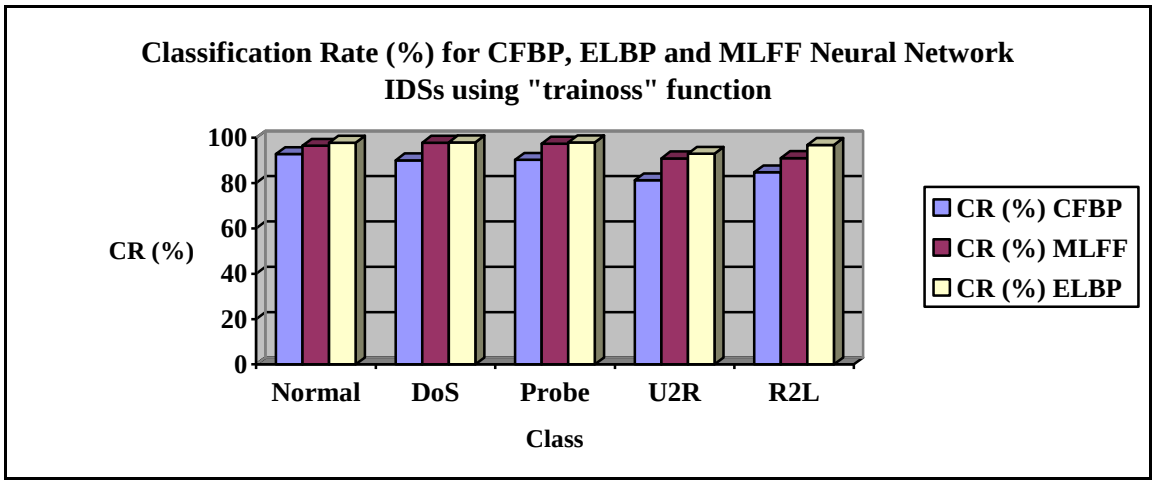

Figure 1 Classification Rate with "trainoss" function

5. Performance using "traingdx" function: Figure 2 shows that the classification rate for various classes of data with MLFF neural network is close to classification rate with ELBP network, but still the performance of ELBP neural network IDS is better.

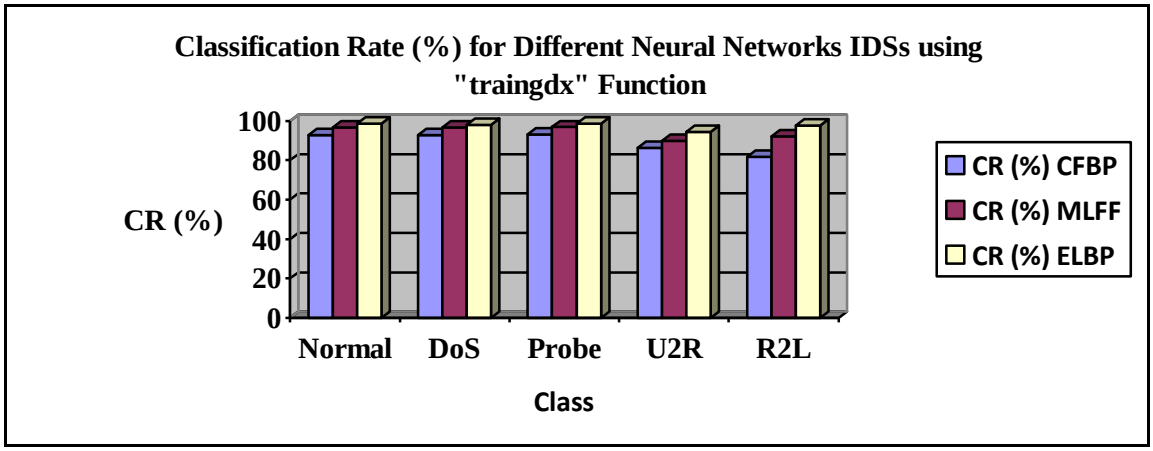

Figure 2 Classification Rate with "traingdx" function

6. Performance of Different Neural Network IDSs using "trainrp" function: Figure 3 shows that the performance of ELBP neural network IDS is better than other two networks with "trainrp" function for all classes of connection records under testing. It is clear from Figure 3 that by using this training function the difference in classification rate compared to all three neural network IDSs is large for all the classes of connections.

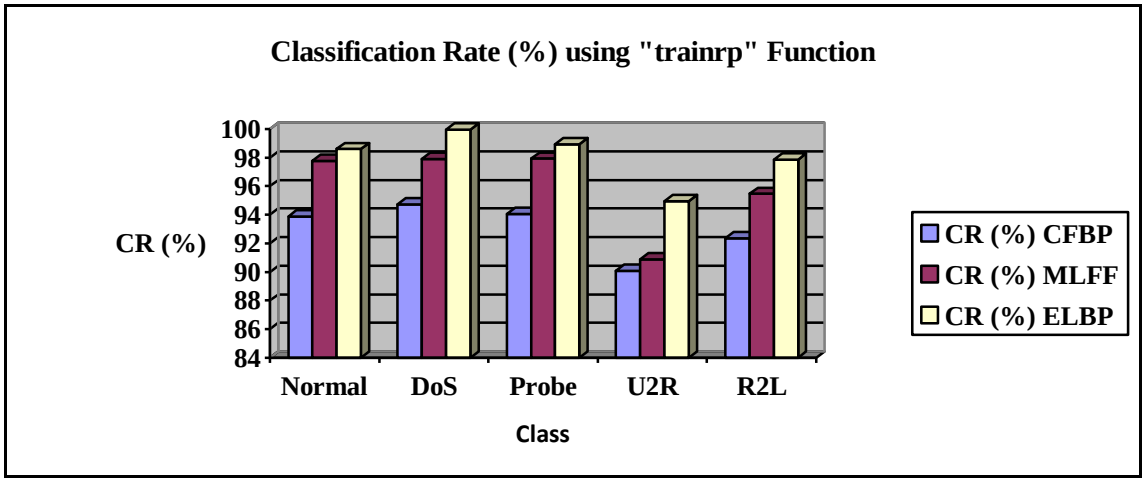

Figure 3 Classification Rate (%) with "trainrp" function

7. Mean Square Error for CFBP, MLFF and ELBP Neural Network IDSs using "trainoss": Smaller mean square error implies better the classification rate and lesser the rate of misclassifications. Table 6 shows that MSE using "trainoss" training function is larger for CFBP neural network (i.e. average of 0.06854) in comparison to other networks.

8. Mean Square Error for CFBP, MLFF, and ELBP Neural Networks IDSs using "traingdx": ELBP neural network shows that the mean square error is 0.0051 (average MSE for all classes), which is smaller than the MSE of 0.058 with CFBP neural network and a MSE of 0.0192 with MLFF neural network (see table 7).

Table 6 MSE using "trainoss" function

Class	MSE for CFBP	MSE for MLFF	MSE for ELBP
Normal	0.0236	0.0092	0.0037
DoS	0.0468	0.0034	0.0035
Probe	0.0447	0.0037	0.0035
U2R	0.1642	0.0447	0.0324
R2L	0.0634	0.0346	0.0084

Table 7 MSE using "traingdx" function

Class	MSE for CFBP	MSE for MLFF	MSE for ELBP
Normal	0.0221	0.0084	0.0029
DoS	0.0234	0.0049	0.0032
Probe	0.0207	0.0046	0.0021
U2R	0.0607	0.0457	0.0136
R2L	0.1631	0.0324	0.0037

9. Mean Square Error for CFBP, MLFF, and ELBP Neural Networks IDSs using "trainrp": Smaller mean square error implies better classification rate and lesser the rate of misclassifications. Table 8 shows that MSE using "trainrp" training function is larger for CFBP neural network in comparison to other networks.

Table 8 MSE using "trainrp" function

Class	MSE for CFBP	MSE for MLFF	MSE for ELBP
Normal	0.0203	0.0036	0.0028
DoS	0.0198	0.0033	0.0005
Probe	0.0121	0.0031	0.0020
U2R	0.0457	0.0442	0.0124
R2L	0.0246	0.0102	0.0034

10. Average MSE Performance of Different Neural Networks: Figure 4(a) shows the variations in MSE with "trainrp" training function are larger for CFBP neural network IDS in comparison to other neural networks IDSs.

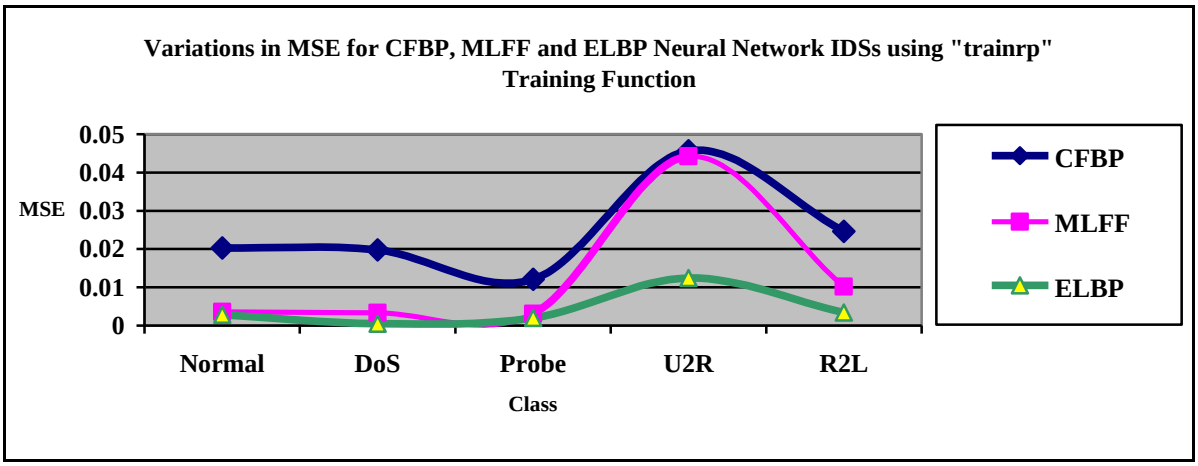

Figure 4(a): MSE using "trainrp" training function

Figure 4(b) shows the variations in MSE with "traingdx" training function is larger for CFBP. However the MSE is very small for the attack types DoS and Probe, with both the ELBP and MLFF neural networks.

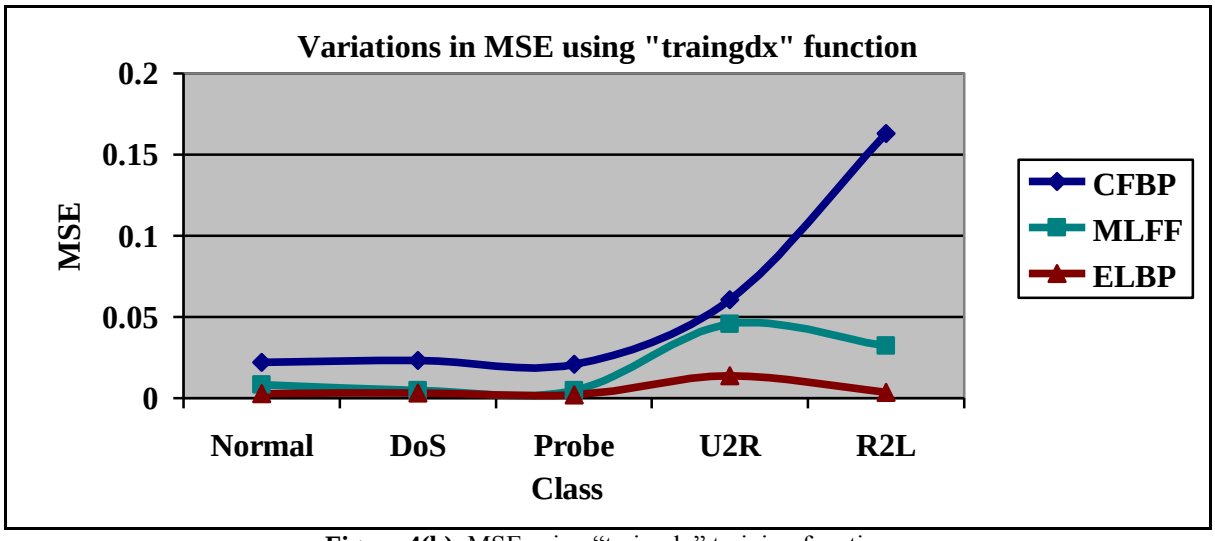

Figure 4(b): MSE using "traingdx" training function

Figure 4(c) shows the variations in MSE with "trainoss" training function is smaller and very similar for both the MLFF and ELBP neural networks for all the attack classes, except for attack R2L for which MLFF neural network IDS has greater value as compared to ELBP.

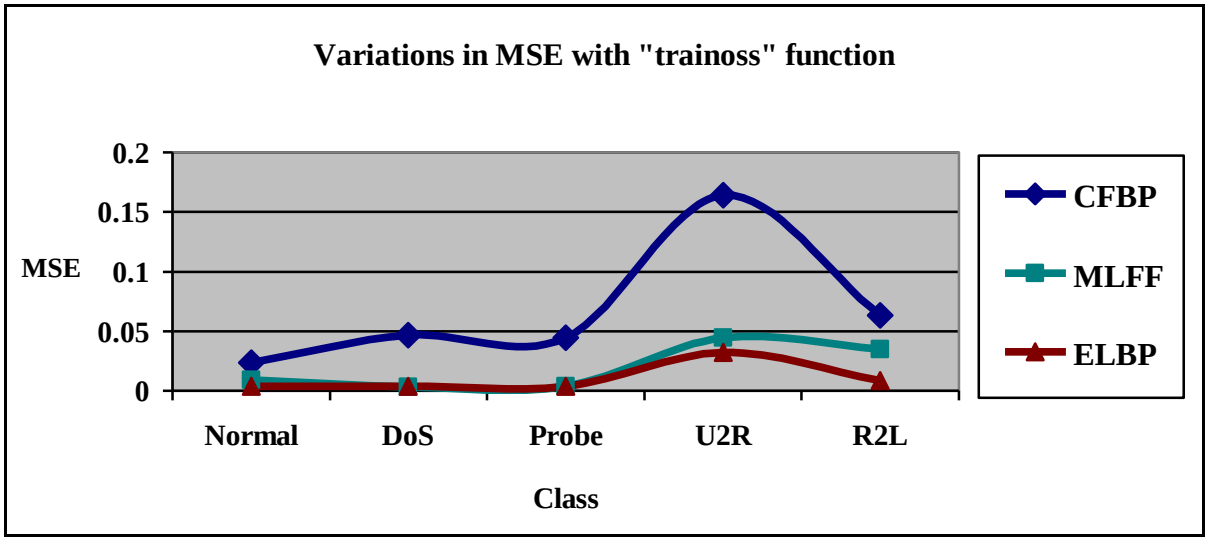

Figure 4(c): MSE using "trainrp" training function

11. Average Classification Rate for Three Different Neural Network Intrusion detection systems: Table 9 shows overall Classification Rate (%) for "41 41 40 1" Neural Network Architectures for CFBP, MLFF and ELBP Intrusion detection systems using all 3 different training functions. Results show that ELBP Neural Network with training function "trainrp" is better of all the other networks with Classification Rate of 98.04 %.

Table 9 Average Classification rate using different training functions

Training Function	CR (%) for CFBP	CR (%) for MLFF	CR (%) for ELBP
Trainoss	87.82	94.50	96.66
Traingdx	89.39	94.69	97.53
Trainrp	93.82	95.98	98.04

Figure 5 shows the variations in classification rate (%) for three different neural networks IDSs with 3 different training functions. Classification rate percentages with ELBP are higher of the other two neural networks. The ELBP neural network intrusion detection system achieves the better performances for all the classes of attacks using "trainrp" training function. The classification rate (%) for MLFF neural network using training function "trainoss" and "trainrp" is almost similar to ELBP IDS.

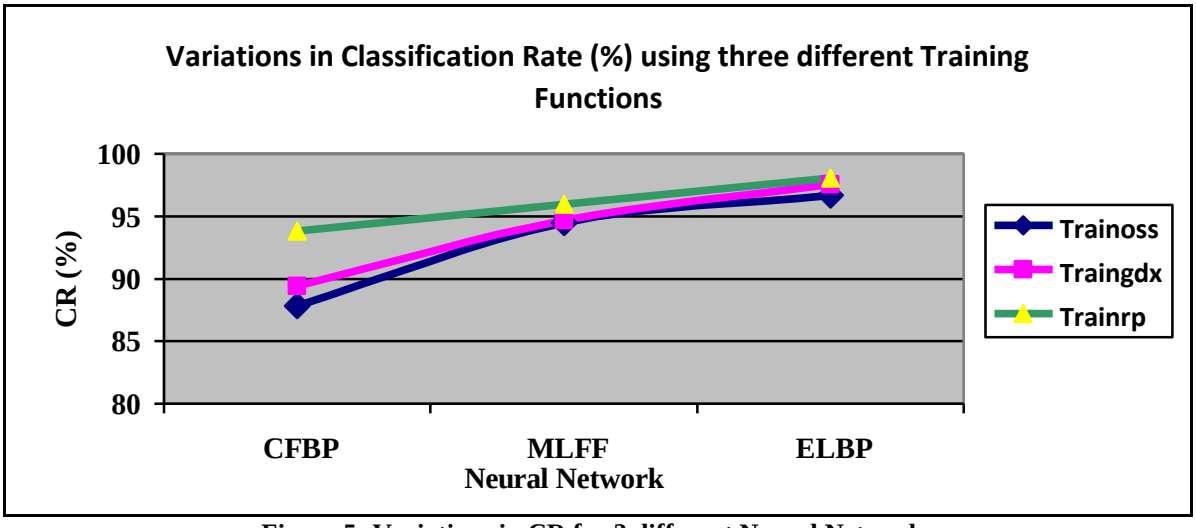

Figure 5: Variations in CR for 3 different Neural Networks

9.2 Test results by varying number of nodes in hidden layer

Here in this section, the results are shown for testing new untrained connection records on the "41 41 35 1" neural architecture. This implies that number of nodes in second hidden layer is changed from 40 to 35. The effects of change in number of nodes on the performances of the MLFF, CFBP and ELBP neural networks are given in this section.

1. **Effects of change in Number of Nodes in second Hidden Layer for CFBP Network:** Table 10 shows, with the change in the number of nodes in the second HL from 40 to 35, the CR degrades significantly thereby leading to an increase in misclassifications.

Table 10 Change in Classification Rate for CFBP with 35 Nodes in Hidden Layer 2

Network Architecture	CR with "trainoss"	CR with "traingdx"	CR with "trainrp"
35 Nodes in HL 2	0.8682	0.8834	0.9287
41 Nodes in HL 2	0.8782	0.8939	0.9382

2. Effects of change in number of nodes in second Hidden Layer for MLFF Network: Table 11 shows, with the change in the number of nodes in the second hidden layer from 40 to 35 the classification rate degrades significantly for MLFF network. For example CR is 0.9498 with 35 nodes using "trainrp", while it is 0.9598 with 40 nodes.

Table 6.11 Change in Classification Rate for MLFF with 35 Nodes in Hidden Layer 2

Network Architecture	Classification rate with "trainoss"	Classification rate with "traingdx"	Classification rate with "trainrp"
35 Nodes in HL 2	0.9421	0.9361	0.9498
41 Nodes in HL 2	0.9450	0.9469	0.9598

3. Effects of change in Number of Nodes in second Hidden Layer for ELBP Network: Table 12 shows degradation in classification rate for ELBP Neural Network with the change in the number of nodes in the second hidden layer from 41 to 35.

Table 12 Change in Classification Rate for ELBP with 35 Nodes in Hidden Layer 2

Network Architecture	Classification rate with "trainoss"	Classification rate with "traingdx"	Classification rate with "trainrp"
35 Nodes in HL 2	0.9566	0.9657	0.9724
41 Nodes in HL 2	0.9666	0.9753	0.9804

4. Average Classification Rate with Three Different Training Function by Varying Nodes of Second Hidden Layer: Table 13 shows classification rate (%) for CFBP, MLFF and ELBP Neural Networks with three different training functions, by varying number of nodes in second hidden layer to 35 from 40. This average of CR is taken as the union of all attack and normal connections recognized correctly by the classifier. The CR achieved through ELBP neural network using training function"trainrp" is higher as compared to other two neural networks by varying nodes in second hidden layer. The overall CR percentage is reduced for all networks as compared to CR given in Table 9.

Table 13 Average Classification rate by varying Number of Nodes in Hidden Layer

Training Function	CR (%) for CFBP	CR (%) for MLFF	CR (%) for ELBP
trainoss	86.82	94.21	95.66
traingdx	88.34	93.61	96.57
trainrp	92.87	94.98	97.24

6. Variations in Average Classification Rate with Change in Number of Nodes: Figure 6 shows graphically the variations in the classification rate for each of the 5 classes shown in table 14, for ELBP network using training function "trainrp", with the change in the number of nodes from 40 to 35 in HL 2. Similar variations exist for other networks.

Table 14(a) Variations in CR for "41 41 40 1" and "41 41 35 1" ELBP Network

Class	CR for ELBP with 40 nodes in HL	CR for ELBP with 35 Nodes in HL2
Normal	0.9842	0.9798
DoS	0.9978	0.9897
Probe	0.9874	0.9850
U2R	0.9478	0.9391
R2L	0.9769	0.9688

Figure 6 shows the changes in CR with the change in the number of nodes in hidden layer.

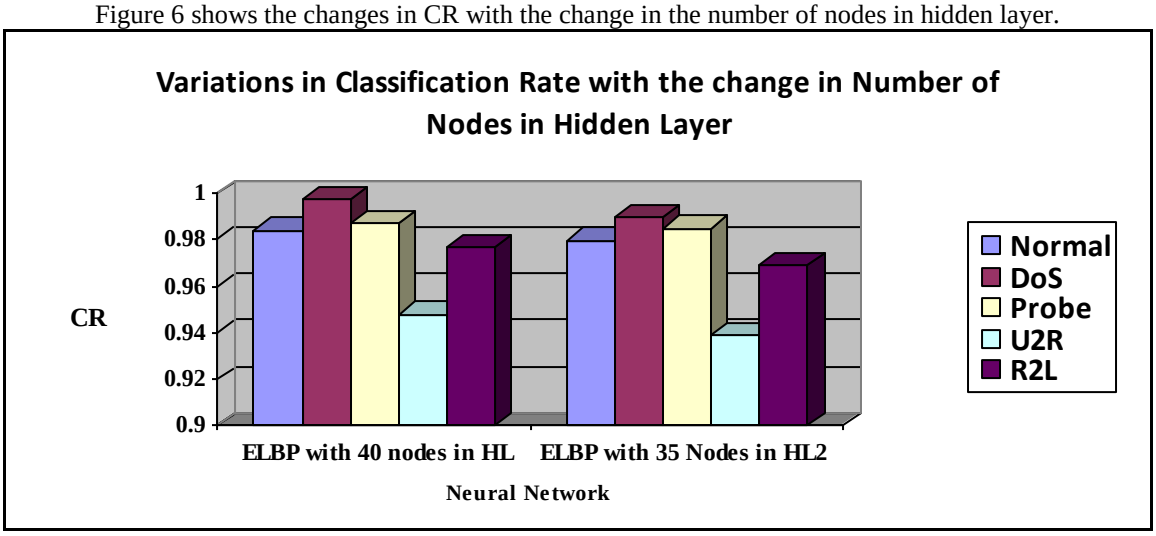

Figure 6: Effects of number of hidden layer nodes on ELBP network

Table 14(b) shows the variations in classification rate for different classes of data with MLFF neural network. The classification rate for attack class U2R is reduced to a greater extent by the change in number of nodes in second hidden layer from 41 to 35. CR for U2R is changed from 90.88 % to 84.78 %.

Table 14(b) Variations in CR for "41 41 40 1" and "41 41 35 1" MLFF Network

Class	CR for MLFF with 40 nodes in HL	CR for MLFF with 35 Nodes in HL2
Normal	0.9775	0.9601
DoS	0.9788	0.9533
Probe	0.9792	0.9671
U2R	0.9088	0.8478
R2L	0.9546	0.9343

Table 14(c) shows the variations in classification rate for different classes of data with CFBP neural network. The classification rate for attack classes U2R and R2L is reduced to a greater extent by the change in number of nodes in second hidden layer from 41 to 35. CR for U2R is changed from 90.08 % to 84.23 %. CR for U2R is changed from 92.36 % to 85.56 %.

Table 14(c) Variations in CR for "41 41 40 1" and "41 41 35 1" CFBP Network

Class	CR for MLFF with 40 nodes in HL	CR for MLFF with 35 Nodes in HL2
Normal	0.9386	0.9123
DoS	0.9473	0.9186
Probe	0.9405	0.9236
U2R	0.9008	0.8423
R2L	0.9236	0.8556

9.3 Results for SOFM Neural Network Intrusion Detection System

Parameters of SOFM Neural Network training are defined in Table 13. Table 15 below shows CR (%) using SOFM for 5 different classes. CR (%) for class "Probe" i.e. 98.81 % is higher of all other classes by using SOFM neural network.

Table 15 CR (%) for SOFM Neural Network

Class	CR (%)
Normal	98.76
DoS	97.72
Probe	98.81
U2R	91.36
R2L	91.36

9.4 Results for TCP Connection Records to Other Services

KDD 99 Dataset consists of connection records collected over TCP connections. Here in this section, we evaluate the performance of TCP connection records to other services which constitute the majority of records of KDD 99 Dataset. The services considered are: "http", "ftp_data", "telnet", "all". Results evaluated on the basis of False Positive Rate (FPR) & FNR.

1. False Positive Rate and False Negative Rate results For TCP to Service Type "all": Service type "all" indicates that all the services considered together that are available in KDD 99 dataset. ELBP Neural network has lowest FPR for Probe i.e. 0.8 and lowest FNR for DoS i.e. 0.10 as shown in Table 16.

Table 16 FPR & FNR for TCP to "all" services

Neural Network	Normal		DoS		Probe		U2R		R2L	
	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR
CFBP	0.33	0.36	0.24	0.25	0.21	0.22	0.49	0.51	0.37	0.11
MLFF	0.22	0.15	0.18	0.11	0.13	0.14	0.18	0.17	0.16	0.15
ELBN	0.11	0.11	0.9	0.10	0.8	0.12	0.15	0.14	0.12	0.12
SOFM	0.19	0.13	0.16	0.11	0.12	0.11	0.17	0.16	0.15	0.13

2. False Positive Rate and False Negative Rate results for TCP to service "http": Table 17 shows the FPR and FNR for the Service type "http". Lowest FPR i.e. 0.07 and FNR i.e. 0.11 for attack type DoS with ELBP neural network. The highest FPR is 0.42 for attack type R2L with CFBP neural network. Also the highest FNR is 0.36 for attack type R2L with CFBP neural network. As discussed before higher the FPR and FNR the greater the number of misclassification for a particular class.

Table 17 FPR & FNR for TCP to "http" services

Neural Network	Normal		DoS		Probe		U2R		R2L	
	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR
CFBP	0.20	0.26	0.19	0.19	0.22	0.23	0.23	0.21	0.42	0.36
MLFF	0.20	0.17	0.18	0.18	0.21	0.17	0.23	0.19	0.20	0.19
ELBN	0.12	0.12	0.07	0.11	0.08	0.11	0.13	0.15	0.15	0.12
SOFM	0.17	0.16	0.15	0.16	0.17	0.13	0.18	0.17	0.16	0.18

3. False Positive Rate and False Negative Rate results for TCP to service "ftp_data": Table 18 shows the FPR and FNR service type "ftp_data" is considered. Lowest FPR is for DoS i.e. 0.07 with ELBP. In terms of false positive rate and false negative rate, SOFM neural network performs better than MLFF neural network and CFBP neural networks for intrusion detection. Like for example, for class U2R the FPR is 0.23, 0.23 and 0.17 with CFBP, MLFF and SOFM neural networks respectively.

Table 18 FPR & FNR for TCP to "ftp_data" services

Neural Network	Normal		DoS		Probe		U2R		R2L	
	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR
CFBP	0.20	0.27	0.19	0.22	0.22	0.14	0.23	0.16	0.42	0.12
MLFF	0.20	0.16	0.18	0.18	0.21	0.18	0.23	0.21	0.20	0.20
ELBN	0.12	0.13	0.07	0.12	0.08	0.23	0.13	0.41	0.15	0.34
SOFM	0.16	0.15	0.15	0.14	0.16	0.20	0.17	0.22	0.17	0.22

4. False Positive Rate and False Negative Rate results for TCP to service "telnet": Table 19 shows the FPR and FNR for service type "telnet" . FPR and FNR are higher for all classes with CFBP neural network; this means that number of misclassifications is more with CFBP neural network intrusion detection system.

Table 19 FPR & FNR for TCP to "telnet" services

Neural Network	Normal		DoS		Probe		U2R		R2L	
	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR
CFBP	0.18	0.20	0.17	0.18	0.20	0.19	0.37	0.32	0.43	0.29
MLFF	0.18	0.17	0.20	0.19	0.17	0.13	0.32	0.22	0.41	0.19
ELBN	0.08	0.13	0.06	0.12	0.07	0.12	0.12	0.17	0.07	0.19
SOFM	0.11	0.15	0.17	0.18	0.15	0.12	0.18	0.19	0.21	0.17

5. Variations in false negative rate with different services for DoS attacks: Figure 7 shows the variations in FNR for four different neural networks IDSs, over different services for DoS attack. Lowest FNR is for ELBP neural network with service type all. Using CFBP neural network for intrusion detection the number of misclassifications is higher than with any other neural network for intrusion detection shown in figure 7.

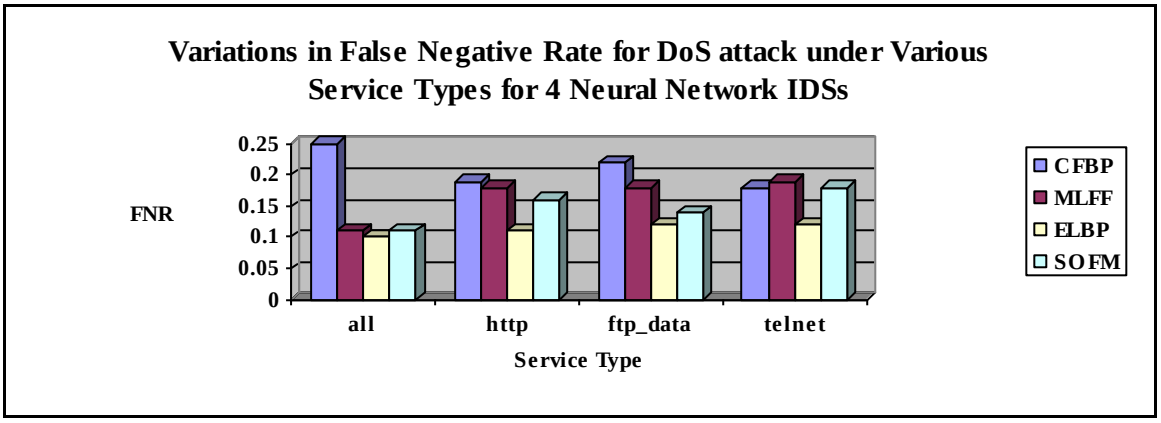

Figure 7: Variations in FNR for attack type "DoS"

6. Variations in false negative rate with different services for Probe attacks: Figure 6.8 shows the variations in FNR for 4 different neural networks over different services for Probe attack. SOFM and ELBP networks show better performance in terms of lower FNR.

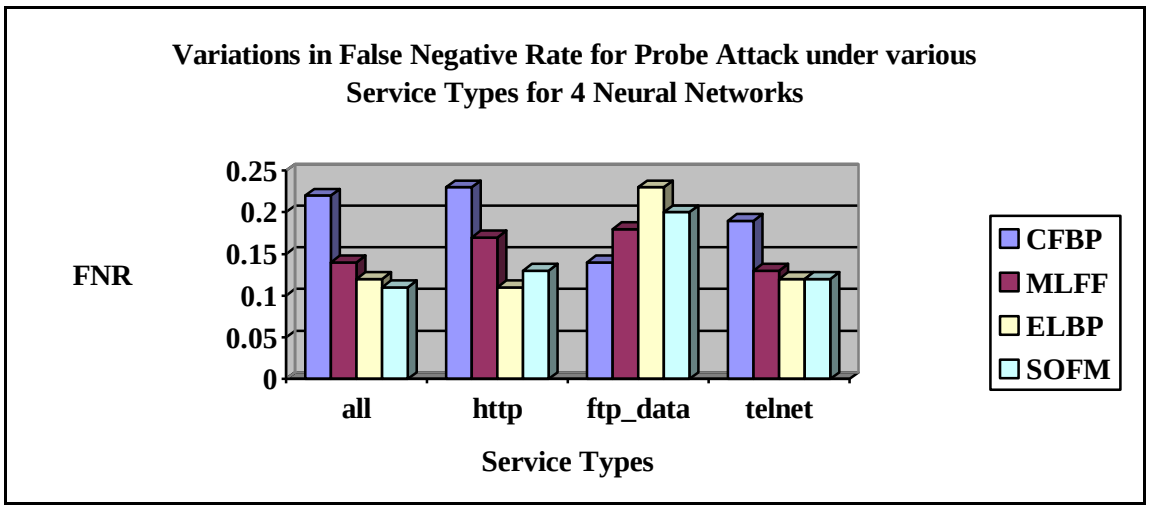

Figure 8: Variations in FNR for attack type "Probe"

7. Variations in false negative rate with different services for U2R attacks: Figure 9(a) shows the variations in FNR for four different neural networks over different services for U2R attack. MLFF neural network shows an average performance for all type of services i.e. FNR is more or less similar for all services.

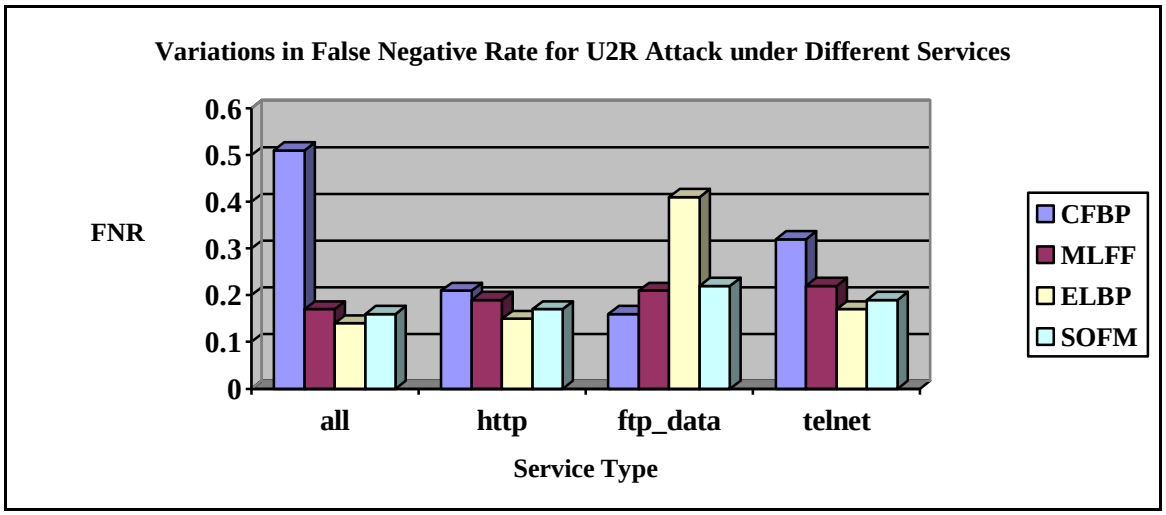

Figure 9(a): Variations in FNR for attack type "U2R"

8. Variations in false negative rate with different services for R2L attacks: Figure 9(b) shows the variations in FNR for four different neural networks over different services for R2L attack. MLFF neural network shows an average performance in terms of FNR for all type of services i.e. FNR is more or less similar for all services.

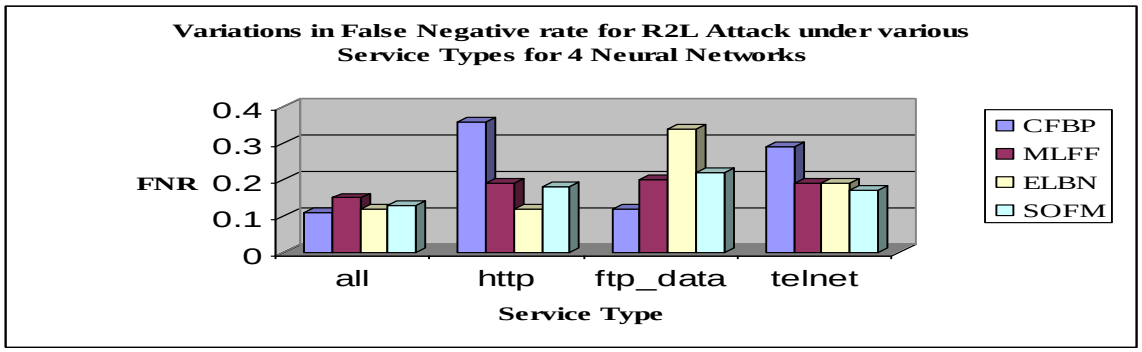

Figure 9(b): Variations in FNR for attack type "R2L"

9.5 Training Time for Different Neural Network IDSs

This section describes the performance of different neural network intrusion detection systems in terms of time required for training. Table 20, 21 and 22 shows the variations in training times for 3 different neural networks IDSs, using 3 different training functions. It can be concluded from these results that ELBP requires least training time as compared to other networks and performs better in terms of training times with "trainrp" training function.

Table 20 Variations in training times with "trainoss" function

Neural Network Architecture	Training Time (sec) for "trainoss"			
	DoS	Probe	U2R	R2L
CFBP	856	869	220	200
MLFF	746	668	211	213
ELBP	578	632	200	209

Table 21 shows that there is great reduction in training time for attack type "DoS" using ELBP neural network with "traingdx". Training time of 631 seconds, 431 seconds and 256 seconds is required by CFBP, MLFF and ELBP neural network respectively for attack type DoS.

Table 21 Variations in training times with "traingdx" function

Neural Network Architecture	Training Time (sec) for "traingdx"			
	DoS	Probe	U2R	R2L
CFBP	631	492	218	219
MLFF	431	393	200	202
ELBP	256	257	195	193

Table 22 Variations in training times with "trainrp" training function

Neural Network Architecture	Training Time (sec) for "trainrp"			
	DoS	Probe	U2R	R2L
CFBP	550	547	197	168
MLFF	293	237	194	190
ELBP	253	257	186	184

Figure 10 shows the variations in training times of three different neural networks with training function "trainoss". With ELBP intrusion detection system, the training time is reduced largely.

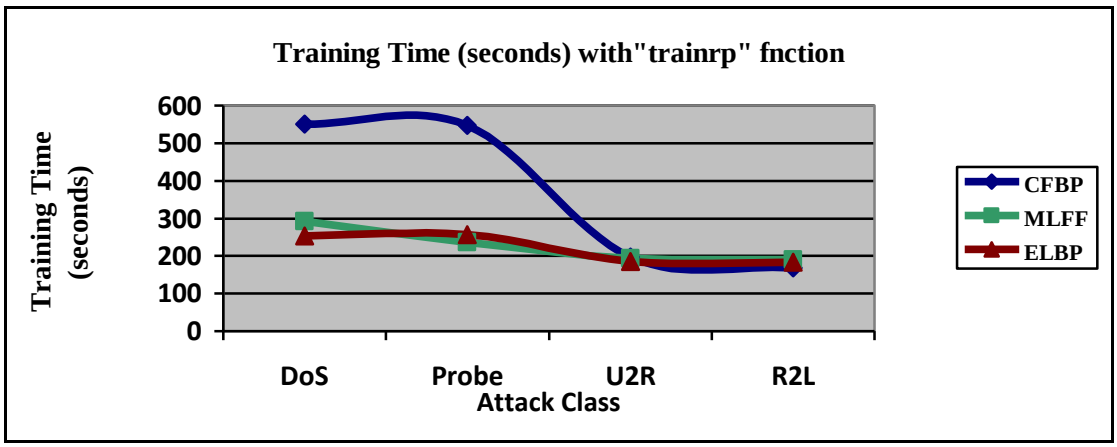

Figure 10: Variations in Training Time with "trainrp"

ELBP neural network give better results in comparison to other networks in terms of Classification rate, False Positive Rate and False Negative Rate. Table 23 shows some of the important finding derived from above in terms of good performances.

Table 23 Important Findings

Parameter	Better Results obtained with
Neural Network Architecture	Elman Back Propagation Neural Network
Training Function	Trainrp
Number of Nodes in Different Layers	41, 41, 40, 1

X. CONCLUSIONS

This paper presented a comparative study of four different neural networks based intrusion detection systems in a network based computer system. The data required for the development of neural network intrusion detection systems have been obtained from KDD Cup' 99 data. Totally 4 category of attacks which include 24 number of intrusion from the computer network were considered in the developed models.

We find that larger the distinct number of records for training the neural network, the better the classification rate for untrained records because the generalization capability of the network is enhanced, thereby leading to better classification rate for different classes of records.

Change in the number of nodes in hidden layer resulted in the change in classification rate and also change in the false positive rate and false negative rate for the neural network based intrusion detection systems. With number of nodes close to the size of input vector (i.e. 41), the classification rate achieved is better than the one achieved by using number of nodes less the size of the input vector.

For all kinds of attacks considered the ELBP intrusion detection system shows very good classification rate, smaller false positive and false negative rates, as compared to the results reported by the simple MLFF, CFBP and SOFM neural network based IDS.

REFERENCES

[1] A.K.Ghosh and A.Schwartzbard, "A study in using neural networks for anomaly and misuse detection", in the Proceedings of USENIX Security Symposium, 1999.

[2] K.Fox,R.Henning,J.Reed,and R.Simonian, "A neural network approach towards intrusion detection", in the Proceedings of the 13th National Computer Security Conference, 1990.

[3] James Cannady and Jim Mahaffey, "The application of Artificial Intelligence to Misuse Detection", in proceedings of the first Recent Advances in Intrusion Detection (RAID Conference) ,1998.

[4] Fox, Kevin L., Henning, Rhonda R., and Reed, Jonathan H., "A Neural Network Approach Towards Intrusion Detection", in the Proceedings of the 13th National Computer Security Conference, 1990.

[5] P.GaneshKumar, D.Devaraj, V.Vasudevan, "Artificial Neural Network for Misuse Detection in Computer Network", in the proceedings of the International Conference on Resource Utilization and Intelligent Systems (INCRUIS-2006), 2006.

[6] DARPA Dataset documentation: http://www.ll.mit.edu/IST/ideval/data/data_index.html .

[7] The MathWorks-MATLAB and Simulink for Technical Computing, MATLAB online help http://www.Mathworks.com/product/matlab/tryit.html .

[8] KDD-cup' 99 dataset, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[9] L. Girardin,"An eye on network intruder-administrator shootouts", in Proceedings of the Workshop on Intrusion Detection and Network Monitoring (ID '99), pages 19 .28, Berkeley, CA, USA, 1999.USENIX Association.

AUTHORS

First Author – Afrah Nazir, Computer Engineering Section, Women's Polytechnic, A. M. U. Aligarh, 202002, India