Volume 4, Issue 2, February 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

To Improve Efficiency of Intrusion Detection System by Using Improved Multilayer Perceptron Algorithm

Richa Sondhiya[*], Mahendra ku Mishra, Manish Shrivastava

Department of IT,
LNCT Bhopal, India

Abstract— ***In last few decades the use of internet has increased rapidly. Simultaneously the growing area of business, research, education has also increased use of computer network. Because of all these, it is necessary to keep computer network very safe and secure. For this purpose intrusion detection system are used, which task is to filter attacks, intrusions which occur in network. Intrusion detection system is a very popular tool to detect attacks. Intrusion Detection Systems monitor computer system to find out sign of security violations over network. When IDSs detects such sign triggers it has to report them to generate the alerts. These alerts tell the user about intruders. The alerts are presented to a human analyst then human analyst evaluates those alerts and initiates an adequate response. In Practice, IDS observe numbers of attacks per day. It also has to deal with different types of attacks. When IDS deals with network intrusions one of the important concerns is to generate true alarms, it means sometime it mistakenly generate an alarm for a legitimate user. Various soft computing techniques are used in Intrusion Detection System. A neural network Technique Multilayer Perceptron algorithm has gained importance in addressing network security issues, these works to classify and identify attacks in network. In this paper we proposed a technique called Improved Multilayer Perceptron which aims to identify attacks with a high detection rate, better time efficiency. We present Improved MLP Technique and compare with Traditional MLP Technique for intrusion detection. These techniques are applied to the KDD Cup99 data set .In addition; a Comparative analysis shows the advantage of Improved MLP Technique over Traditional MLP Technique over detection rate and time efficiency. Our main aim is to improving efficiency of network intrusion detection system by using improved Multilayer Perceptron Algorithm.***

Keywords—: Intrusion Detection System, Neural Network, Multilayer Perceptron, Improved Multilayer Perceptron algorithm.

I. INTRODUCTION

The very fast growth of development of internet and World Wide Web has changed the scenario of internet and networking technologies. This rapid growth of internet and network technologies has also increased the number of users and amount of data which travel in network. In another hand the chances of data loss, hacking and intrusion is also increased. In order to this the demand of network security techniques has also increased to keep the data and network resources secure. To provide security to data and resources various techniques has been proposed one of them is Intrusion Detection System [14]. An intrusion detection system (IDS) is a component of the network security environment. Its main task is to identify between suspicious or intrusive activity and normal behaviour. The aim of intrusion detection is to build a system which itself watch network activity and detect such intrusive attacks. Once an attack is recognized, the system administrator is informed who takes adequate action to deal with the intrusion.

The IDS works with two main approaches one is misuse detection and another is anomaly detection. Misuse detection approach is based on stored signature pattern. In this a database contains signatures and if data does not match with pre stored data it is counted as intrusion. Anomaly detection approach based on behaviour pattern if data traffics behavior varies with a normal user behaviour then it is counted as intruders. IDS is host-based (HIDS), network based (NIDS) or a combination of both types that is Hybrid Intrusion Detection System. HIDS usually observes logs and system calls on a single system, in another hand a NIDS typically observes traffic and data flows and Network data packets on a network segment, and that's why it checks number of hosts in a very short events. Generally, one use to work with very large amount of network data, because of this it is very hard and tiresome work to classify the records manually in order to detect a correct intrusion. Labelled data can be obtained by simulating intrusions, but this will be limited only for the set of known attacks. That's why new types of attacks which occur in future cannot be handled, if those were not part of the training data. Sometimes with manual classification, we can only able to identifying the previously known (at classification time) types of attacks, that's why we restrict our detection system to identify only those types of attacks. To solve these deficiencies, we need a technique to detect intrusions when our training data is unlabeled, as well as for detecting new and un-known types of attacks. A method that offers reliability in this task is anomaly detection. Anomaly detection finds anomalies in the data (i.e. data instances in the data that deviate from normal or regular ones). It also makes us able to detect new types of intrusions, because these new types will, by assumption, be deviations from the normal network usage. It is very difficult and sometimes not possible, to detect malicious users when misfeasor is a person who uses is authorized user to use the network and who uses the network in a legitimate way. For example, there is probably not a very reliable way to know whether someone who appropriately logged into a system and work as the intended user of that specific system, or at the situation if the password was stolen.

Under all these assumptions and consideration we design a system which created clusters from its input data, and then it itself form labelled clusters as containing either normal or attacks type of data instances, and finally used these clusters to classify network data instances as either normal or intrusive. Both the training and testing was done using 10% KDDCup'99 dataset [2], which is a very popular and most frequently used intrusion attack dataset. Most of the clustering techniques assume a well-defined differentiation between the clusters so that each pattern can only belong to one cluster at a time. This supposition can neglect the natural ability of object which existing in numbers of clusters. For this reason and with the aid of fuzzy reasoning, fuzzy clustering can be employed to overcome the deficiencies. The membership of a pattern in a given cluster can vary between 0 and 1. In this model a data object belongs to the clusters where it has the highest membership value.

II. RELATED WORK

Some IDS designers take ANN as a pattern recognition method. Pattern recognition is implemented by using a feed-forward neural network which is trained according to need. At the time of training training, the neural network parameters are optimized so that they can be associated with the outputs (each output represents a class of computer network. Security in IDS is a very important area of research. Connections, like normal and attack) with corresponding input patterns (here every input pattern is represented by a feature vector which is extracted from the characteristics of the network connection record). The neural network is used to identify the input pattern and try to output the corresponding class. When a connection record which does not have any output associated with it is treated to give as an input in network, the neural network provides the output that belongs to an already taught input pattern that is least different from the given pattern [6]. The most commonly designed application of neural networks in IDSs is that it train the neural network upon a sequence of information units, among them each of which may be an audit record or a sequence of commands. The input to the net consists of the current command and the past *w* commands (*w* is the size of window of commands under examination). Once the net is trained on a set of representative command sequences of a user, it constitutes (learns) the profile of the user and when it is put in action, it can find out the variance of the user from its profile [4], [6]. Usually recurrent neural networks are used frequently for this work.

Ryan et al. [3] described an off-line anomaly detection system (NNID) which utilized a back propagation MLP neural network. The MLP was trained to identify users' profile and at the end of each log session, the MLP evaluated the users' commands for possible intrusions (offline).

The authors explained their research in a small computer network with ten users. Each feature vector described the connections of a single user for a whole day. 100 most important commands are used to describe a user's behavior. They also used a 3 layer MLP (with 2 hidden layers). The MLP identified the user correctly in 22 cases out of 24.

Canady [2] used a three layer neural network for offline classification of connection records in normal and misuse classes. The system designed in this study was intended to work as a standalone system (not as a preliminary classifier whose result may be used in a rule-based system). The feature vector used in [2] was composed of nine features all describing the current connection and the commands used in it. A dataset of 10,000 connection records including 1,000 simulated attacks was used. The training set included 30% of the data. The final result is a two class classifier that succeeded in classification of normal and attack records in 89-91% of the cases. In yet another study [10], the authors used three and four layer neural networks and reported results of about 99.25% correct classification for their two class (normal and attack) problem.

Cunningham and Lippmann [11] used ANNs in misuse detection. They used an MLP to detect Unix-host attacks by searching for attack specific keywords in the network traffic. Different groups used self-organizing maps (SOM) for intrusion detection [5].

Brian Hay et. al [6] have focused on data authentication, data integrity, querying and outsourcing the encrypted data. Their research says that, the risks can arise at operational trust modes, resource sharing, new attack strategies and digital forensics. In operational trust modes, the encrypted communication channels are used for cloud storage and do the computation on encrypted data which is called as homomorphism encryption. New attack strategies like Virtual Machine Introspection (VMI) can be used at virtualization layer to process and alter the data. The issues are clarified using the digital forensics techniques namely the ephemeral nature of cloud resources and seizing a "system" for examination. In most of the earlier studies [2], [3], [10], the neural networks were implemented systems with two

Possible outputs: normal or anomaly. In all of those studies, both types of records were inserted like some types of attacks and a set of normal records were included in the dataset; however, the output of the neural network was 1 or 0 for normal or attack conditions (the attack type was not determined by the neural network). The present study is designed to solve a multi class problem in which not only the attack records are differentiated from normal ones, but also the attack type is identified.

III. MULTILAYER PERCEPTRON ALGORITHM

This section presents the design of a feed word neural network that's accustomed compress image within the analysis works. Multilayer-perceptron formula may be a wide used learning formula in Artificial Neural Networks. The Feed-Forward Neural specification is capable of approximating most issues with high accuracy and generalization ability. This formula is predicated on the error-correction learning rule. Error propagation consists of 2 passes through the various layers of the network, a passing play, and a backward pass. Within the passing play the input vector is applied to the sensory nodes of the network and its result propagates through the network layer by layer. Finally a collection of outputs is made because the actual response of the network. Throughout the passing play the junction weight of the networks area unit all mounted. Throughout the rear pass the junction weights area unit all adjusted in accordance with AN error-correction rule. The particular response of the network is deducted from the required response to supply a slip-up signal. This error signal is then propagated backward through the network against the direction of Synaptic conditions. The junction weights area unit adjusted to form the particular response of the network move nearer to the required response.

Algorithm

The algorithm for Perceptron Training works according to the back-propagation algorithm which is discussed previously. This algorithm can be coded in any programming language, and in the case of our work we have coded this by using the use of the sigmoid function **f(net)**. This is because it has a simple derivative.

Algorithm:

1. **Initialize the values of weights and threshold.**
   Set all weights and thresholds values to small random values.
2. **Present the input and desired output value**
   Present input values as $\boldsymbol{X_p = x_0 ,x_1 ,x_2 ,...,x_{n-1}}$ and target output as $\boldsymbol{T_p = t_0 ,t_1 ,...,t_{m-1}}$ where n is the number of input nodes and m is the number of output nodes. Set $\mathbf{w_0}$ to be **-ø**, the bias, and $\mathbf{x_0}$ to be always 1. For pattern association, $\boldsymbol{X_p}$ and $\boldsymbol{T_p}$ represent the patterns to be associated. For classification, $\boldsymbol{T_p}$ is set to zero except for one element set to 1 that corresponds to the class that $\boldsymbol{X_p}$ is in.
3. **Calculate the actual output value**
   Each layer calculates the following:
   $y_{pj} = f\,[w_0x_0 + w_1x_1 + ... + w_nx_n]$
   This is then forwarded to the next layer as an input. The final layer outputs values $\mathbf{o_{pj}}$.
4. **Adapts weights**
   Start from the output we now work backwards.
   $w_{ij}(t+1) = w_{ij}(t) + ñþ_{pj}o_{pj}$, where **ñ** is a gain term and $þ_{pj}$ is an error term for pattern **p** on node **j**.
5. **For output units**
   $þ_{pj} = ko_{pj}(1 - o_{pj})(t - o_{pj})$
   For hidden units
   $þ_{pj} = ko_{pj}(1 - o_{pj})[(þ_{p0}w_{j0} + þ_{p1}w_{j1} + ... + þ_{pk}w_{jk})]$
   where the sum(in the [brackets]) is over the **k** nodes in the layer above node **j**.

Step 5: Use simple majority of the category of nearest neighbours as the prediction value of the new sample

IV. PROPOSED APPROACH

We propose a Neural Network based techniques for network intrusion detection. We have applied few rules to enhance the performance of the algorithm and we have given it's a name called Improved Multilayer Perceptron Algorithm Our main aim To Improving Efficiency of Network Intrusion Detection System By Using Improved Multilayer Perceptron Algorithm.

Improved Multilayer Perceptron Algorithm

In this section, we have described the new algorithm IMLP will be which it would improve the performance of the MLP algorithm. The convergence speed of the learning process can be improved significantly by IMLP through adjusting the error, which will be transmitted backward from the output layer to each unit in the intermediate layer. In MLP Algorithm, the error at a single output unit is defined as:

$$\delta^0{}_{pk} = (Y_{pk} - O_{pk}).$$

Where the subscript "P "refers to the pth training vector, and "K "refers to the kth output unit. In this case, Ypk is the desired output value, and Opk is the actual output from kth unit, then δpk will propagate backward to update the output-layer weights and the hidden-layer weights. In this proposed work error is propagated back but before propagating that error we will perform the following calculation on error value. So the error at a single output unit IMLP will be as:

$$New\delta^0{}_{pk} = (1 + e^{(Y_{pk} - O_{pk})^2})$$

$$, if(Y_{pk} - O_{pk}) \geq zero.$$

$$New\delta^0{}_{pk} = -(1 + e^{(Y_{pk} - O_{pk})^2})$$

$$, if(Y_{pk} - O_{pk}) < zero.$$

Where New δ˚pk is considered as the new error value in the IMLP algorithm. Improved MLP Algorithm uses two forms of Newδ˚pk, one is positive and another is negative. We can see it check the error upon two rules and according to the satisfied rule it calculate the new error value by applying exponential formula on it. It uses exponential function because the exponential function always returns zero or positive values (and the weight adapts operation for many output units, need to decrease the actual outputs rather than increasing it). This New δ˚pk will minimize the errors of each output unit more quickly than the old δ˚pk, and the weights on certain units weight change very large from their starting values.

Steps of an Improved MLP Algorithm

1. Apply the input example to the input units.
2. Calculate the net-input values to the hidden layer units.
3. Calculate the outputs from the hidden layer.
4. Calculate the net-input values to the output layer units.
5. Calculate the outputs from the output units.
6. Calculate the error term for the output units, but replace Newδ˚pk with δ˚pk.
7. Calculate the error term for the output units, using Newδ˚pk, also.
8. Update weights on the output layer.
9. Update weights on the hidden layer.
10. Repeat steps from step 1 to step 9 until the error (Ypk – Opk) is acceptably small for each training vector pairs.

V. EVALUATION MEASURES

To evaluate the performance of the MLP algorithm, there are numbers of performance measure like time efficiency, detection rate, False Positive, True Positive etc. In our proposed work we have taken time efficiency as a performance measure and also calculated detection rate to better compare the performance of the system.

Time Efficiency – The time efficiency of the algorithm is measured by the time that algorithm takes for classification of the record. The records are classified into two categories that is normal and attacks.

Detection Rate - (also known as Detection Rate or Completeness): It is defined as rate of detecting the attacks. It is calculated as a ratio of the intrusion found and total numbers of records and events or total no of intrusions.

VI. EXPERIMENTAL ANALYSIS

The experimental analysis of both algorithms is performed on the basis of their time efficiency of classifying the attacks. We have taken KDDcup99 dataset which have both the types of record in dataset. Both the algorithm we used to apply on dataset a check that which algorithm performs the classification faster. We have also calculated the detection rate of the algorithms to check the performance of the system. The Detection rate is calculated by their rate of detecting true attacks for a given specific interval of time. We can also check true positive, which determines the percentage of truly detected intrusions. It finds that how many attacks are correctly triggered. The main goal of our proposed work is that it finds and enhances the learning ability of IDS (Intrusion Detection System) in less time. Here we want to improve the performance of IDS by using IMLP in place of Traditional MLP algorithm. Here IMLP Algorithm is contrasted to a clustering k-mean algorithm which use the full set of trials which are taken from the KDD Cup99 dataset, It has 5000 samples. The genuine dataset has 5 million notes. These are based on diverse attacks in which 1% experiments of about 5000 notes which we have used in our trial. Each records available in KDDcup99 facts and figures set, represents a distinct connection between two networks owner, which is comprised by couple of well-defined mesh protocols. Every connection has 41 characteristics for representation. It contains the rudimentary characteristics of one-by-one of TCP Connections, the content features, No. of byte, figurers of moved byte etc. In column 2 characteristics of KDDcup99 Data set is there which conveyed byte, flag is. We are focusing anomaly detection with supervised Learning algorithm. That's why all the records which are tagged as strike are considered as intrusion, and in other hand residual notes was treated as usual. The marks are only utilized for assessing the detection performance of the algorithm. These are not utilized throughout classification process. The classification method which we have utilized in our comparative study has a limitation that it will not handle categorical facts and figures and the categorical characteristic like flag in the data set are changed utilizing 1-to-N Encoding technique Our main aim To Improving Efficiency of Network Intrusion Detection System By Using Improved Multilayer Perceptron Algorithm

. In this experiment we find out classification time of IMLP algorithm and MLP algorithm. At the time of training we classify training data. When training is over we will give a label to each group. It is performed according to the majority type of data in that group like if in any group has majority of data is intrusive then we will label that group as intrusive. Here we have compare the result for hundred clusters which is given in table 1. After comparing the result of both the algorithm we find that IMLP execution time is less than MLP. The data set only contain numeric values.

.

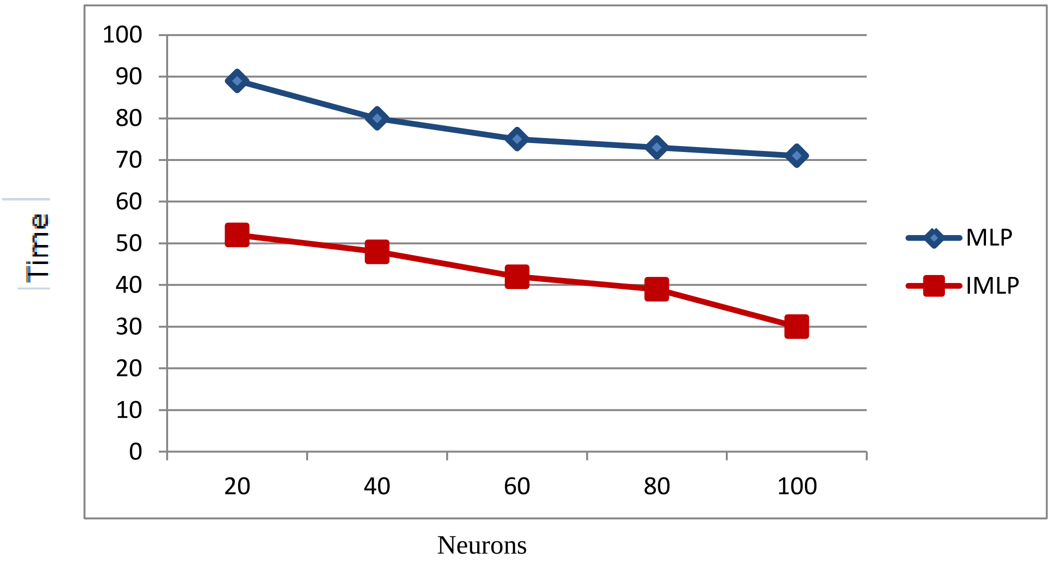
TABLE 1

Time efficiency of both the algorithm with 100 neurons

Neurons	**Algorithm**	
	MLP **IMLP**	

	Time(Ms)
	Time(ms)	

100	89	52
80	80	48
60	75	42
40	73	39
20	71	30



Figure(1): Time Efficiency between MLP and IMLP Algorithm

Experiment 2

Now we find the detection rate of MLP and IMLP Algorithm. To evaluate the accuracy of a system, we use two different indicators, which were used in: Detection Rate (DR) and False Alarm Rate (FAR). In our proposed system we have calculated DR. DR equals the number of intrusions which is divided by the total number of intrusions in the data set.

$$\text{DR} = \frac{\text{the number of intrusions}}{\text{The total number of intrusions in the data set}}$$

We partitioned 5000 instances of KDD-99 data using the IMLP Algorithm and MLP algorithm with different initial values of k. The Detection rate of IMLP algorithm and MLP are

TABLE 2:
SUMMARY DETECTION RESULTS WITH 100 NEURONS

Neurons	**IMLP Algorithm**	**MLP Algorithm**
20	89	64
40	92	68
60	94	70
80	95	76
100	97	81

The table (2) shows that MLP algorithm has low detection rate than IMLP Algorithm. A algorithm is used in intrusion detection system is so good when it has low false positive rate and high Detection rate. IMLP Algorithm map reduce the false positive and it has high detection rate for detect the unseen or new attack. The graph for the IMLP and MLP Algorithm algorithms are omitted for comprehensibility and better visualization, particularly because they are visibly worse Our main aim To Improving Efficiency of Network Intrusion Detection System By Using Improved Multilayer Perceptron Algorithm

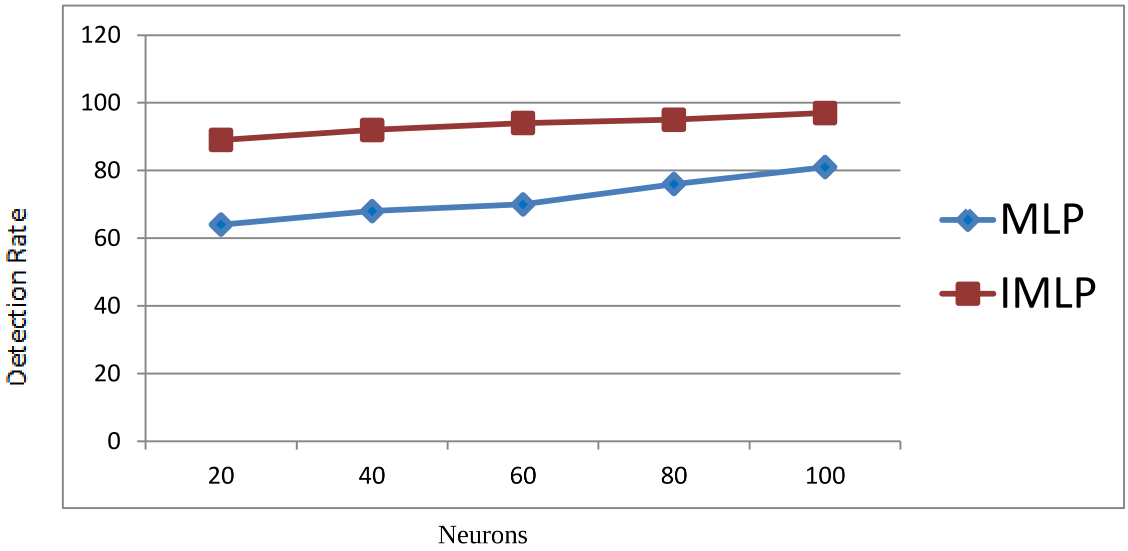


Figure 2: Detection Rate of the IMLP Algorithms and MLP Algorithm

Figure (2) Show the Graph for detection rate of the MLP algorithms and IMLP Algorithm with 100 Neurons.

VII. CONCLUSION

An approach for a neural network based intrusion detection system, is used to classify the normal and attack pattern, has been presented in this paper. We have applied Improved MLP method which gives us better time efficiency as compared to traditional MLP. Then we have also calculated detection rate to find the accuracy and after comparing the results given in table 1 and table2 we can say that Improved IMLP performed better as compared to traditional MLP in both the measures. At the same time it increased the generalization capability of the neural network and decreased the training time. It must be taken into concern that the long training time of the neural network was most of the time arise because of to the huge number of training vectors of computation facilities. However, when the neural network parameters were determined by training time, classification of a single record is done in a very short span of time.

In this paper we introduce Improved MLP in IDS to protect user. This algorithm is can be easily scalable by allowing different format of data to apply into this algorithm IMLP for more reliable IDS solution. The implemented IMLP is just a step to improve the performance of the existing system. An interesting future topic is the implementation is Improved MLP with different activation function. To practically implement this technique deployment, performance and scalability issues need to be considered as the next step. The implemented Improved MLP can be a great step in the application area of MLP like pattern recognition, face recognition etc. This Technique can also be used to be implemented in grid and cloud computing environment.

REFRENCES

[1] Mehdi MORADI and Mohammad ZULKERNINE "A Neural Network Based System for Intrusion Detection and Classification of Attacks" 978-1-4673-0309-5/12/$31.00 ©2012 IEEE

[2] Application of Artificial Neural Network in Detection of Probing Attacks 2009 IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009), October 4-6, 2009, Kuala Lumpur, Malaysia

[3] D. E. Denning, "An intrusion detection model," *IEEE Transactions on Software Engineering*, vol. 13, no. 2, pp. 222– 232, 1987.

[4] James Cannady, "Artificial neural networks for misuse detection," Proceedings of the 1998 National Information Systems Security Conference (NISSC'98), Arlington, VA, 1998.

[5] J. Ryan, M. Lin, and R. Miikkulainen, "Intrusion Detection with Neural Networks," *AI Approaches to Fraud Detection and Risk Management: Papers from the 1997 AAAI Workshop*, Providence, RI, pp. 72-79, 1997.

[6] K. Fox, R. Henning, J. Reed, and R. Simonian, "A neural network approach towards intrusion detection," Proceedings of 13th National Computer Security Conference, Baltimore, MD, pp. 125-134, 1990.

[7] P. Lichodzijewski, A.N. Zincir Heywood, and M. I. Heywood, "Host-based intrusion detection using self-organizing maps," *Proceedings of the 2002 IEEE World Congress on Computational Intelligence*, Honolulu, HI, pp. 1714-1719, 2002.

[8] H. Debar, M. Becker, and D. Siboni, "A neural network component for an intrusion detection system," Proceedings of 1992 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, California, pp. 240 – 250, 1992.

[9] Daivid Poole, Alan Makworth, and Randi Goebel, Computational Intelligence, New York: Oxford University Press, 1998.

[10] Sergios Theodorios and Konstantinos Koutroumbas, *Pattern Recognition*, Cambridge: Academic Press, 1999.

[11] Kristopher Kendall, “A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems,” *Masters Thesis, MIT,* 1999.

[12] Srinivas Mukkamala, “Intrusion detection using neural networks and support vector machine,” *Proceedings of the 2002 IEEE International* Honolulu, HI, 2002.

[13] R. Cunningham and R. Lippmann, “Improving intrusion detection performance using keyword selection and neural networks,” *Proceedings of the International Symposium on Recent Advances in Intrusion Detection,* Purdue, IN, 1999.

[14] MATLAB online support: www.mathworks.com/access/helpdesk/help/techdoc/matlab.sht ml.

[15] C. Sinclair, L. Pierce, and S. Matzner, “An application of machine learning to network intrusion detection,” Proceedings of 15th Annual Computer Security Applications Conference (ACSAC '99), Phoenix, AZ, pp. 371-377, 1999.

[16] R. A. Kemmerer and G. Vigna, “Intrusion detection: a brief history and overview,” Computer, vol. 35, no. 4, pp. 27–30, 2002.

[17] Piero P. Bonissone, “Soft computing: the convergence of emerging reasoning technologies,” *Soft Computing Journal*, vol.1, no. 1, pp. 6-18, Springer-Verlag 1997.

[18] Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks International Journal of Artificial Intelligence And Expert Systems (IJAE), Volume (1): Issue (4)