ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 5, July 2012

Network Weight Updating Method for Intrusion Detection Using Artificial Neural Networks

DR.S.SARAVANAKUMAR[1], JERRIN SIMLA.A[2], L.MEGALAN LEO[3], DR.A.REGARAJAN [4]

Professor Dept of Information Technology, Panimalar Institute of Technology ,Chennai[1]

Department of CSE Asst.Professor Panimalar Institute of Technology ,Chennai[2]

Department of ECE Asst.Professor St.Joseph College of Technology ,Chennai[3]

Department of Computer Science Engineering, Veltech Multitech SRS engineering college, chennai[4]

saravanakumars81@rediffmail.com[1], jerrinsimla@gmail.com[2], megalanleo@gmail.com[3]

Abstract: Application of Artificial Neural Networks (ANN) to intrusion detection has been considered in this research work. Experimental data were collected using KDD database.Using the data collected, the training patterns and test patterns are obtained. An ANN has been used to train the data offline. The weight updating algorithms developed for the ANN are based on the back propagation algorithms, echo state neural network and the functional update method. The method of presenting the patterns to the input layer of the network has been analyzed. The different methods of presenting the input patterns, such as reducing the dimension of the input patterns by a transformation and preprocessing of the input patterns for non linear classifiers have been investigated.In order to find the optimum number of nodes required in the hidden layer of an ANN a method has been proposed, based on the change in the mean squared error dynamically, during the successive sets of iterations. Two classification and four classification problems for training an ANN at different levels have been studied. Optimal discriminant plane technique which is a classical method has been used to reduce the dimension of the input pattern and then used to train an ANN. This reduces the size of the network and the computational effort is reduced drastically.The training of an ANN has been considered by splitting the single configuration into two configurations. The convergence rates of the split network are faster than that of single configuration network.The input patterns are preprocessed and presented to the input layer of ANN. Various types of preprocessing of the patterns are investigated. Comparisons of the classification performance and computational effort of the different weight updating algorithms with different training methods have been given. Several algorithms developed in this research work may find other application areas and in security with little modifications.

Keywords : ANN, weight updating, input layer, network, KDD

I. INTRODUCTION

A single intrusion of a computer network can result in the loss or unauthorized utilization or modification of large amounts of data and cause users to question the reliability of all of the information on the network. There are numerous methods of responding to a network intrusion, but they all require the accurate and timely identification of the attack. Increasing attempts to compromise computer systems by methods ranging anywhere from masquerading as a privileged user to coordinating distributed attack probes across a network have led to increased research in intrusion detection. Intrusion detection systems detect Denial of Service (DoS) attacks either by using a prior knowledge of the types of known attacks or by recognizing deviations from normal system behaviors.

1.1.1 Classification of Attack Detection

- **Attack/Invasion detection**: Tries to detect unauthorized access by outsiders.

- **Misuse Detection**: Tries to detect misuse by insiders, e.g., users that try to access services on the internet by passing security directives. Misuse detection uses a prior knowledge on intrusions and tries to detect attacks based on specific patterns of known attacks.

- **Anomaly Detection**: Tries to detect abnormal states within a network.

- **Host Intrusion Detection System (HIDS):** HIDS works on information available on a system. It can easily detect attacks by insiders as modification of files, illegal access to files and installation of Trojans.

- **Network Intrusion Detection System (NIDS):** NIDS works on information provided by the network mainly packets sniffed from the network layer. It uses protocol decoding, heuristical analysis and statistical anomaly analysis. NIDS detects DoS with buffer overflow attacks, invalid packets, attacks on application layer and spoofing attacks.

II. Approaches to Intrusion Detection

All current intrusion detection systems make four assumptions about the systems that they are designed to protect

a) Activities taken by system users either authorized or unauthorized can be monitored.

b) It is possible to identify those actions which are indications of an attack on a system.

c) Information obtained from the intrusion detection system can be utilized to enhance the overall security of the network.

d) A fourth element which is desirable from any intrusion detection mechanism is the ability of the system to make an analysis of an attack in real time. This would allow the intrusion detection mechanism to limit the adverse effects which are perpetrated on the system. An effective use of this element is probably the most difficult component of an intrusion detection system to achieve. While metrics can be developed which monitor all aspects of a user behavior. The resulting degradation on the overall performance of the system may require that a thorough analysis be conducted off-line, thus eliminating a real-time detection capability. Other approaches, such as pattern recognition are attempting to identify new methods of identifying information system attacks. Wei et al (2009) describes the primary ways an intruder can get into the system is through primary intrusion, system intrusion and remote intrusion.

III. ARTIFICIAL NEURAL NETWORK

An Artificial Neural Network is an abstract simulation of a real nervous system that contains a collection of neuron units communicating with each other via axon connections. Such a model bears a strong resemblance to axons and dendrites in a nervous system. Due to this self-organizing and adaptive nature, the model offers potentially a new parallel processing paradigm. This model could be more robust and user friendly than the traditional approaches. ANN can be viewed as computing elements, simulating the structure and function of the biological neural network. These networks are expected to solve the problems in a manner which is different from conventional mapping. Neural networks are used to mimic the operational details of the human brain in a computer. Neural networks are made of artificial neurons which are actually simplified versions of the natural neurons that occur in the human brain. It is hoped that it would be possible to replicate some of the desirable features of the human brain by constructing networks that consist of a large number of neurons. A neural architecture comprises massively parallel adaptive elements with interconnection networks which are structured hierarchically.

Artificial neural networks are computing elements which are based on the structure and function of the biological neurons. These networks have nodes or neurons, which are described by difference or differential equations. The nodes are interconnected layer wise or intra-connected among themselves. Each node in the successive layer receives the inner product of synaptic weights with the outputs of the nodes in the previous layer. The inner product is called the activation value. The activation value is passed through a non-linear function.

When the vectors are binary or bipolar, hard-limiting non-linearity is used. When the vectors are analog a squashed function is used. Some of the squashed functions are sigmoid (0 to 1), tanh (-1 to +1), Gaussian, logarithmic and exponential. A network with two states of a neuron 0 or 1 and –1 or 1 is called discrete and the same with a continuous output is called analog. In a discrete network at a particular time the state of every neuron is updated, the network is said to be synchronous. If the state of only one neuron is updated, the network is said to be asynchronous. A network is feed forward, if there is no closed chain of dependence among neural states. The same network is feed backward, if there is such a closed chain. When the output of the network depends upon the current input the network is static. If the output of the network depends upon past inputs or outputs, the network is dynamic. If the interconnection among neurons changes with time, the network is adaptive. The synaptic weight updation of the networks can be carried out by supervised methods or by unsupervised methods or by fixed weight association networks methods. In the case of the supervised methods, inputs and outputs are used in the unsupervised methods, only the inputs are used and in the fixed weight association networks methods, inputs and outputs are used along with pre-computed and pre-stored weights. Some of the supervised learning algorithms are the perceptrons, decision based neural networks, adaptive linear element (ADALINE), multi layer perceptrons, temporal dynamic models and hidden Markov analysis. The various unsupervised learning algorithms are neo-cognition, self-organizing feature map, competitive learning, adaptive resonance theory (ART) and the principal component analysis.

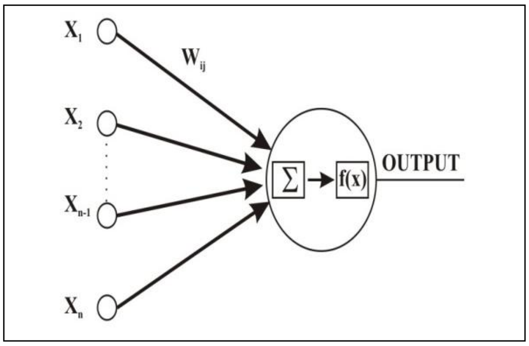

Fig.1.1 Operation of a Neuron

The fixed weight networks are Hamming net, Hopfield net and the combinatorial optimization. The total pattern recognition system constitutes instantiation space, feature extraction, training the network, and the testing the network. Rumelhart et al (2008) has made the concept of ANN widely known through their publications.

IV. AREAS OF APPLICATIONS OF ARTIFICIAL NEURAL NETWORKS

The application domains of ANN are in association / clustering / classification, pattern recognition, regression / generalization and in optimization. The different areas of applications are in computer vision, signal/image processing, speech / character recognition, remote sensing and in controls. Recent applications of ANN include the machining processes, process planning, inventory control, shape based classification, machine fault diagnosis and cold forging.

V. WEIGHT UPDATION METHODS

The neural network approach does not require any modeling between process parameters and process state variables. The network maps the input domains on to output domains. The inputs are packet parameters and the outputs are virus information. The combination of input and output constitutes a pattern. Many patterns will be called data. During training of ANN, the network learns the training patterns by a weight updating algorithm. The training of ANN is stopped when a desired performance index of the network is reached. The weights obtained at this stage are considered as final weights. During implementation of ANN for intrusion detection, the data coming from the network are transformed with the full weights obtained during the training of ANN. Every output of the network is checked. If the outputs are within the desired values detection is enabled.

Training process

Packet information is extracted. The unimportant information is removed. The information is further stored as patterns to create a data file. The data file contains huge patterns. The patterns are further processed to extract representative patterns that can be used for training the ANN with specific ANN algorithm. At the end of training a final weight matrix is obtained which is stored in a file.

Testing process

Incoming packets are stripped of the information and any irrelevant information is filtered. The information is further processed with the final weights to obtain a value in the output layer of the ANN. Based on the output obtained further classification of the presence of any intrusion is done. Based on the type of intrusion action is initiated.

VI. COLLECTION OF DATA

The KDD 99 intrusion detection datasets are collected based on the DARPA initiative which provides designers of intrusion detection systems (IDS) with a benchmark on which to evaluate different methodologies. A simulation is made of a factitious military network consisting of three target machines running various operating systems and services. Additional three machines are then used to spoof different IP addresses to generate traffic. A sniffer records all network traffic using the transmission control protocol dump format. The total simulated period is seven weeks. Normal connections are created to profile that expected in a military network and attacks fall into one of four categories.

Denial of Service: Attacker tries to prevent legitimate users from using a service.

Remote to Local: Attacker does not have an account on the victim machine, hence tries to gain access.

User to Root: Attacker has local access to the victim machine and tries to gain super user privileges.

Probe: Attacker tries to gain information about the target host.

In 2001, the original transmission control protocol dump files were preprocessed for utilization in IDS benchmark of the International Knowledge Discovery (IKD) and Data Mining Tools Competition. Packet information in the transmission control protocol dump file is summarized into connections. Specifically a connection is a sequence of the transmission control protocol packets starting and ending at some well defined times between which data flows from a source IP address to a target IP address under some well defined protocol. This process is completed using IDS resulting in 13 features for each connection. List of features with their descriptions and data types are in Table 3.2.

VII. Categories of Features

Features are grouped into four categories, listed below:

Basic Features: Basic features can be derived from packet headers without inspecting the payload.

Content Features: Domain knowledge is used to assess the payload of the original TCP packets. This includes features such as the number of failed login attempts.

Time-based Traffic Features: The time within which an intrusion takes place repeatedly is used as feature. This feature takes into account the frequency of intrusion as features.

Host-based Traffic Features: The features are based on attacks collected for more than a specified time period. Host based features are therefore designed to assess attacks, which span intervals longer than 2 seconds. In IKD and Data Mining Tools Competition, only “10% KDD” dataset is employed for the purpose of training. 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. Because of their nature, denial of service attacks account for the majority of the dataset. On the other hand the “Corrected KDD” dataset provides a dataset with different statistical distributions than either “10% KDD” or “Whole KDD” and contains 14 additional attacks. The list of class labels and their corresponding categories for “10% KDD” are detailed in Table 3.1. Sample intrusion patterns are given in Table 3.3.

Dataset	DoS	Probe	U2r	R2l	normal
Corrected	229853	4166	70	16347	60593

Table 3.1 Basic characteristics of the KDD99 intrusion detection datasets in terms of number of samples

VIII. NORMALIZATION OF THE PATTERNS

The patterns are normalized so that the values of the features are in the range of 0 to 1 and the computational complexity is reduced. The normalization of the patterns is done by

$$x_i = x_i / x_{max} \quad (3.1)$$

where
x_i is the value of a feature, and
x_{max} is the maximum value of the feature.

Feature	Description
duration	Duration of the connection
protocol type	Connection protocol (e.g. tcp, udp)
service	Destination service (e.g. telnet, ftp)
flag	Status flag of the connection
source bytes	Bytes sent from source to destination
destination bytes	Bytes sent from destination to source
land	1 if connection is from/to the same host/port else 0
wrong fragment	Number of wrong fragments
urgent	Number of urgent packets
Hot	Number of hot indicators
Failed logins	Number of failed logins
logged in	1 if successfully logged in else 0
compromised	Number of compromised conditions

Table 3.2 List of features with their descriptions and data types

A. *Selection of patterns for training*

The numbers of classes, which are based on the classification range of the outputs are decided. If only one output is considered the range of classification is simple. If more than one output is considered a combination criterion has to be considered. The total number of patterns is decided for each class. Out of these patterns, the number of patterns to be used for training the network is decided.

The remaining patterns are used for testing the classification performance of the network. The patterns selected for training the network should be, such that they represent the entire population of the data.

The selection of patterns is done using equation 3.2.

$$E_i^2 = \frac{\sum_{j=1}^{nf}(x_{ij} - \bar{x}_j)^2}{\sigma_i^2}$$

where

E_i^2 is the maximum variance of a pattern,

nf is the number of features

$$\sigma_i^2 = \frac{\sum_{j=1}^{nf}(x_{ij} - \bar{x}_j)^2}{L} \quad (3.3)$$

$\bar{x}$ mean for each feature

L total number of patterns

The value of E_i^2 is found for each pattern. Patterns with maximum E_i^2 are chosen from each class for training the network.

IX. TRAINING STRATEGIES FOR THE NETWORK

Hush and Horne (2007) in their work have been presented the latest developments in supervised learning. For the network to learn the patterns different weight updating algorithms have been developed. They are called supervised methods and unsupervised methods. Since both the inputs and outputs are considered for intrusion detection supervised learning technique has been used. The present work involves modification of existing weight updation algorithm, combination of classical method with neural network, method of training the network for more number of patterns, and training the network properly for more than two classifications. The performances of the different methods developed and trained have been compared with the performance of Back Propagation Algorithm. Besides XOR problem has been considered as the standard problem to know the influence of various learning parameters on the training methods developed and in which only one output has been taken into consideration for proper analysis of network behavior. Moreover the number of patterns in each class which are based upon the type of attack has been taken into consideration. To start with only two classes are considered for the purpose of intrusion detection classification. The classification performance of the network for the test patterns with two classifications and four classifications will be analyzed and presented. When more than two classifications are to be considered the training procedure of the network has to be improved.

The network functions on a supervised learning strategy. The inputs of a pattern are presented. The output of the network obtained in the output layer is compared with the desired output of the pattern. The difference between the calculated output of the network and the desired output is called Mean squared error (MSE) of the network for the pattern presented. This error is propagated backwards such that the weights connecting the different layers are updated. By this process the MSE of the network for the pattern presented is minimized. This procedure is summed up. After presenting the last training pattern, the network is considered to have learnt all the training patterns through iterations, but the MSE is large. To minimize MSE the network has to be presented with all the training patterns many times. There is no guarantee that the network will reach the global minimum. Instead it will reach one of the local minima. The MSE may increase which means divergence rather than convergence. Sometimes there may be oscillations between convergence and divergence.

The training of the network can be stopped, either by considering MSE or by considering classification performance as the criterion. When classification performance is considered as the criterion, test patterns are presented at the end of each iteration. Once the desired performance is obtained, training of the network is stopped. When MSE is considered as the criterion, one may not know the exact MSE to which the network has to be trained. If the network is trained till it reaches a very low MSE over fitting of the network occurs. Over fitting represents the loss of generality of the network. That is the network classifies only the patterns which are used during training and not the test patterns.

X. BACK PROPAGATION ALGORITHM

A new improvement of Back Propagation Neural Network learning algorithms with Adaptive gain discussed by Nazri Mohd Nawi (2010). The BPA uses the steepest-descent method to reach a global minimum. The flow-chart of the BPA is given in Figure 3.2. The number of layers and number of nodes in the hidden layers are decided. The connections between nodes are initialized with random weights. A pattern from the training set is presented in the input layer of the network and the error at the output layer is calculated. The error is propagated backwards towards the input layer and the weights are updated. This procedure is repeated for all the training patterns. At the end of each iteration, test patterns are presented to ANN and the classification performance of ANN is evaluated. Further training of ANN is continued till the desired classification performance is reached.

STEPS

FORWARD PROPAGATION

- The weights and thresholds of the network are initialized.
- The inputs and outputs of a pattern are presented to the network.
- The output of each node in the successive layers is calculated.

  o(output of a node) = $1/(1+\exp(\sum w_{ij}\, x_i + \Theta))$
- The error of a pattern is calculated

  $E(p) = (1/2) \sum(d(p) - o(p))^2$

REVERSE PROPAGATION

- The error for the nodes in the output layer is calculated

  δ(output layer) = o(1-o)(d-o)
- The weights between output layer and hidden layer are updated

  $W(n+1) = W(n) + \eta\delta$(output layer) o(hidden layer)
- The error for the nodes in the hidden layer is calculated

  δ(Hidden layer) = o(1-o) $\sum\delta$(output layer) W(updated weights between hidden and output layer)
- The weights between hidden and input layer are updated.

  $W(n+1) = W(n) + \eta\delta$(hidden layer) o(input layer)

The above steps complete one weight updation

- Second pattern is presented and the above steps are followed for the second weight updation.
- When all the training patterns are presented, a cycle of iteration or epoch is completed.
- The errors of all the training patterns are calculated and displayed on the monitor as the mean squared error.

  $E(MSE) = \sum E(p)$

1.Simulation Results

The XOR problem is given in Table 3.4. In this table the output has been modified from 0 to 0.1 and from 1 to 0.9. Because the sigmoid function used the network will never reach either 0 or 1, due to the presence of the exponential function. In this chapter and in the subsequent chapters MSE is fixed as 0.01 whenever XOR problem is used to train the network. There is no sanctity in considering MSE as 0.01 one can consider some other value. The initialization of the weights and the thresholds are in the range of 0.25 to 0.47.

The iterations required by the network which are trained by using BPA for different number of nodes in the hidden layer to reach MSE of 0.01, are shown in Figure 3.3.

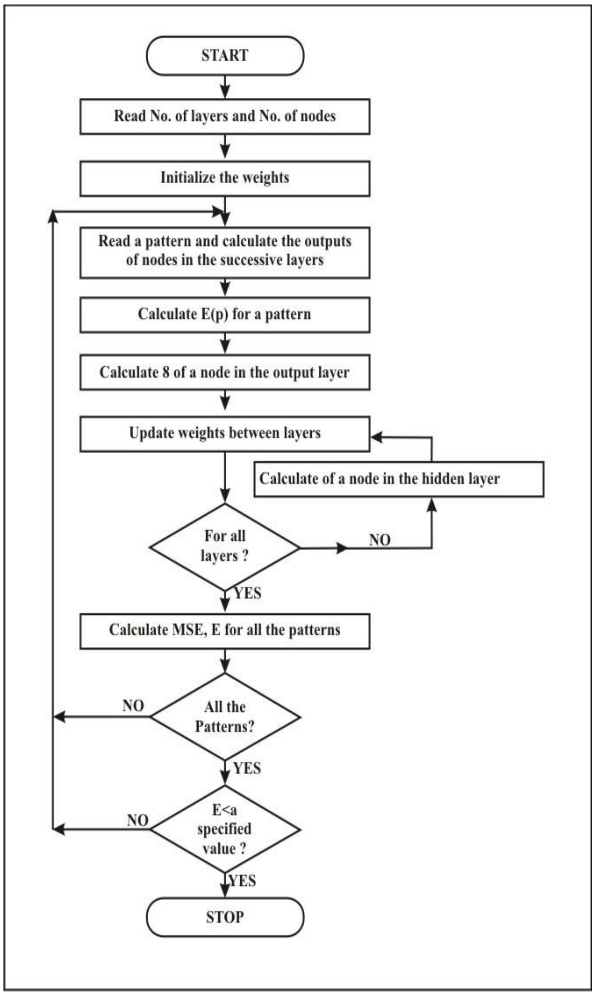
START
Read No. of layers and No. of nodes
Initialize the weights
Read a pattern and calculate the outputs of nodes in the successive layers
Calculate E(p) for a pattern
Calculate 8 of a node in the output layer
Update weights between layers
Calculate of a node in the hidden layer
For all layers ?
NO
YES
Calculate MSE, E for all the patterns
All the Patterns?
NO
YES
E<a specified value ?
NO
YES
STOP

Fig. 3.2 Flow chart of the back propagation algorithm

Pattern no.	Inputs		Output Original Modified	
1	1	1	0	0.1
2	0	0	0	0.1
3	0	1	1	0.9
4	1	0	1	0.9

Table 3.3 XOR problem

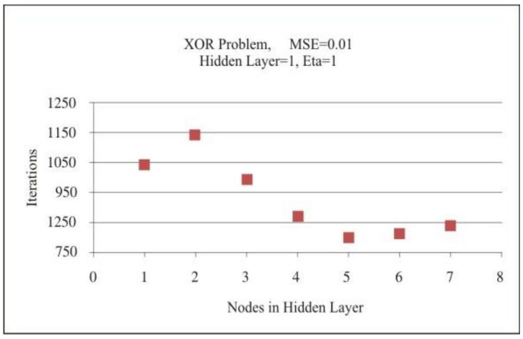

Fig. 3.3 Effect of number of nodes in the hidden layer for the network trained by using Back propagation algorithm

The network is trained with two hidden layers. The total numbers of nodes used in both the hidden layers are 14. The different combinations of number of nodes in the first hidden layer and in the second hidden layer are given in Table 3.5. The convergence rates of the network with two hidden layers and the convergence rates of the network with one hidden layer are shown in Figure 3.4. When there is only one hidden layer and the number of hidden nodes is 6, it requires 985 iterations for the network to reach MSE of 0.01. When there are two hidden layers with 6 nodes in the first hidden layer and 8 nodes in the second hidden layer, it requires 8593 iterations for the network to reach MSE of 0.01. Since it requires more number of iterations for the network with more than one hidden layer, it is sufficient to have only one hidden layer.

No. of nodes in layer 1	No. of nodes in layer 2	No. of iterations to reach MSE of 0.01
4	10	10205
5	9	6851
6	8	8593
7	7	7481
8	6	8593
9	5	9583
10	4	12440

Table 3.4 Number of nodes in each hidden layer

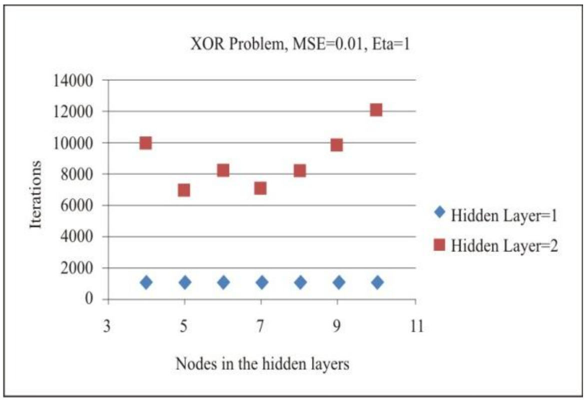

Fig. 3.4 Effect of number of layers in the network trained by using Back propagation algorithm

The learning factor η is supposed to guide the convergence rates of the network to the desired MSE with less number of iterations. It so happens, that sometimes η will make the network to converge to the desired MSE after an increased number of iterations. For 6 nodes in the hidden layer it requires 985 iterations for the network to reach MSE of 0.01 when η is 1.0 and 2404 iterations for the network to reach MSE of 0.01 when η is 0.05. The convergence rates of the network for various numbers of nodes in the hidden layer for different values of η is shown in Figure 3.5.

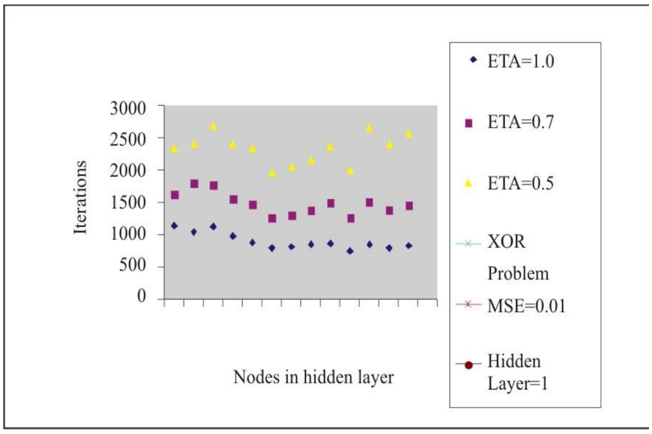

Fig. 3.5 Effect of η in the network trained by using Back propagation algorithm

The network is trained with threshold (θ) and without θ when θ is used updation of θ is done similar to weight updation. The parameter θ is used in all the layers except in the input layer. For 6 nodes in the hidden layer it requires 985 iterations for the network to reach MSE of 0.01 without θ and 1722 iterations for the network to reach MSE of 0.01 with θ. The convergence rates of the network with θ and without θ are shown in Figure 3.6. From the graph it can be seen that network trained without θ converges faster than the network trained with θ.

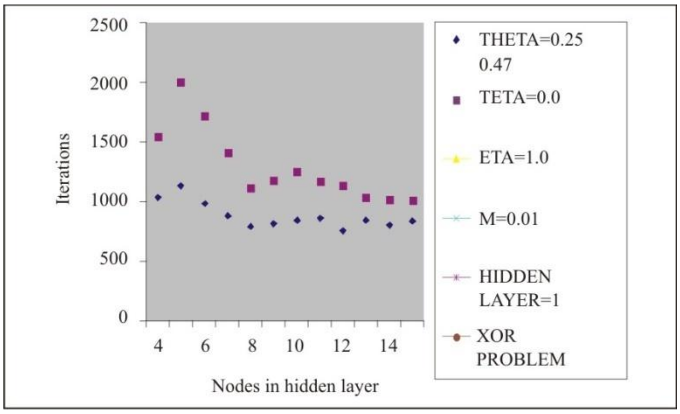

Fig. 3.6 Effect of θ in the network trained by using Back propagation algorithm

To achieve faster convergence of the network an accelerating factor is used which is a parameter called momentum factor (α). The network is trained with α and without α. The value of α is from 0 to 1. For 6 nodes in the hidden layer, it requires 985 iterations for the network to reach MSE of 0.01 without α, and 380 iterations for the network to reach MSE of 0.01 with α. The value of α used is 0.8. For other values of α, the network requires very large number of iterations to reach MSE of 0.01. The convergence rates of the network trained with α and without α are shown on Figure 3.7. The network trained with α requires less number of iterations to reach the desired MSE.

The topology of ANN used the number of nodes in the input layer is 6, the number of nodes in the hidden layer is 2 and the number of nodes in the output layer is 1. The labeling is set as 0.1(normal) and 0.2(attack). It is mandatory to use huge amount of patterns to be presented for training ANN. However it would take enormous amount of time for the ANN to learn the patterns. Hence only 1000 patterns have been considered for training purpose. The dataset has been separated as training and testing. Training indicates the formation of final weights which indicate a thorough learning of intrusion and normal packets along with corresponding labeling.

Figure 3.8 shows the convergence curve for the topology of 2 nodes in the hidden layer with maximum intrusion identification of 91%.

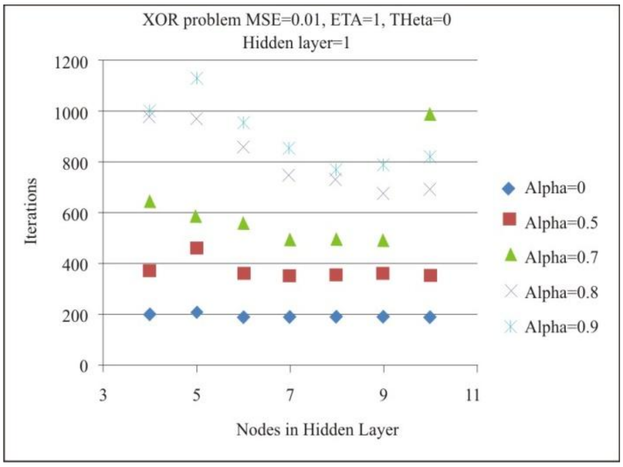

Fig. 3.7 Effect of α in the network trained by using Back propagation algorithm

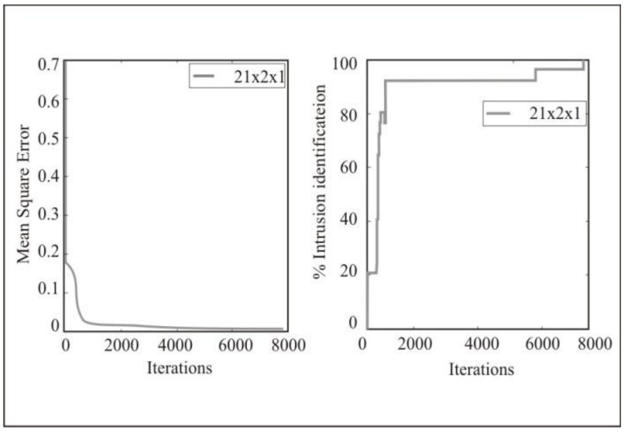

Fig. 3.8 Mean Squared Error of the network trained by using Back propagation algorithm for the intrusion detection

The network was trained to reach MSE of 0.01853. It converged in 405 iterations. If the number of patterns is further increased from 1000 then the convergence iterations will increase. In addition the convergence depends on the orthogonality of data presented for training. If the subsequent patterns are not orthogonal, the convergence would talk a long time or it may not converge.

XI. CONCLUSION

Intrusion detection by using artificial neural network has been considered as the research problem of the exiting conventional methods. The main reason to use artificial neural network (ANN) for intrusion detection is its model free nature. The analysis of the network's performance was carried out by presenting all the test patterns at the end of each iteration. Further training was stopped when the performance of the network reached maximum classification of the test patterns.

REFERENCES

[1] S. Saravanakumar, Amruth Kumar.A, Anandaraj.s and s.Gowtham , Algorithms Based on Artificial Neural Networks for Intrusion Detection in Heavy Traffic Computer Networks ,published in International Proceedings of Computer Science and Information Technology *,vol.20 (2011)*

[2] S. Saravanakumar, Umamaheshwari, D.Jayalakshmi, R.Sugumar ,Development and Implementation of Artificial Neural Networks for Intrusion Detection in Computer Network, Published in Vol. 10 of international journal of computer science and network security

[3] Dr. S. Saravanakumar, T.A. Mohanaprakash, Ms. R. Dharani and Dr. C. Jaya Kumar , Analysis of ANN-based Echo State Network Intrusion. Detection in Computer Networks Published in **International** Journal of Computer Science and Telecommunications [Volume 3, Issue 4, April 2012]

[4] S. Saravanakumar, Jaya Kumar and S. Purushothaman, Design and Implementation of Echo State Network for Intrusion Detection, published in Journal of Computing, Volume 3, Issue 9, September 2011.

[5] Sampada Chavan, Khusbu Shah, Neha Dave and Sanghamitra Mukherjee, 2004, "Adaptive Neuro-Fuzzy Intrusion Detection Systems", Proceedings of the International Conference on Information Technology: Codingand Computing (ITCC'04)

[6] Gang Kou, Yi Peng, Yong Shi, And Zhengxin Chen, (2006), "Network Intrusion Detection by Multi-group Mathematical Programming based Classifier", Sixth IEEE International Conference on Data Mining -Workshops (ICDMW'06)

[7] Helman, P. and Liepins, G., (1993), "Statistical foundations of audit trail analysis for the detection of computermisuse", IEEE Transaction on Software Engineering, 19(9):886-901

[8] Kohonen T., 1990, "The self-organizing map", Proceedings of the IEEE, Vol. 78, No. 9, pp. 1464-1480

[9] Lippmann R. P., 1987, “An introduction to computing with neural nets”, IEE Transactions On Acoustics, Speech and Signal Processing Magazine, Vol. 35, No. 4, pp. 4-22

[10] Hush D. R., and Horne B. G., 1993, “Progress in supervised Neural networks”, IEEE Signal ProcessingMagazine, Vol. 10, No. 1, pp. 8-39.

[11] Mix. D. F, R. A. Jones, 1982, “A Dimensionality Reduction Techniques Based on a Least Squared Error Criterion”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 4, No. 1, pp. 537 – 544.

[12] Jaeger H, “The echo state approach to analyzing and training recurrent neural networks”, (Tech. Rep. No. 148). Bremen: German National Research Center for Information Technology, 2001.

13] Jaeger, H.; Tutorial on training recurrent neural networks, covering BPPT, RTRL,EKF and the “echo state network” approach (Tech. Rep. No. 159).; Bremen: German National Research Center for Information Technology, 2002.

[14] Kosuke Imamura, Kris Smith, 2004, “Potential Application of Training Based Computation to Intrusion Detection”, 25-29 July, Budapest, Hungary

[15] Meng Joo Er. Shiqian Wu, Juwei Lu and Hock Lye Toh, “Face Recognition with Radial Basis Function (RBF) Neural Networks,” IEEE Trans. on Neural Networks, Vol. 13, No.3, pp. 697 – 910, May 2002

[16] Moses Garuba, Chunmei Liu, and Duane Fraites, 2008, Intrusion Techniques: Comparative Study of NetworkIntrusion Detection Systems, Fifth InternationalConference on Information Technology: New

[17] Thomas Holz, 2004, “An Efficient Distributed Intrusion Detection Scheme”, Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC’04)

[18] Wei Hu and Weiming Hu, 2005, “Network-based Intrusion Detection Using Adaboost Algorithm”, Proceedingsof the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI’05).*ferential Geometry*. CRE Press, 1998.

Biography

S.SARAVANAKUMAR
has more than 11 years of teaching and research experience. He did his Postgraduate in ME in Computer Science and Engineering at Bharath engineering college, chennai, and Ph.D in Computer Science and Engineering at Bharath University, Chennai. He occupied various positions as Lecturer, Senior Lecturer, Assistant Professor, Associate Professor and Professor & HOD. He has published more than 20 research papers in High Impact factor International Journal, National and International conferences and visited many countries like Taiwan, Bangok and Singapore. He has guiding a number of research scholars in the area Adhoc Network, ANN, Security in Sensor Networks, Mobile Database and Data Mining under Bharath University Chennai, Sathayabama University

MEGALAN LEO.L has more 4 years experience in teaching . He did post graduate in ME in ece at Sathayabama university, Chennai, and under graduate in ECE at Dr. sivanthi Adithanar college,Tiruchendur from Anna University. He has published more than 3 papers in various national and international conferences. He is currently working has Asst. Professor in St.Joseph college of Engineering and technology, Chennai. His area interest in Artificial Intelligence and Network security, operating system .

JERRIN SIMLA A
Has 5 years experience in teaching . He did post graduate in ME in computer science engineering at Sathayabama university, Chennai, and under graduate in CSE at St. Xavier Catholic college from Anna University. He has published more than 5 papers in various national and international conferences. He is currently working has Asst.Professor in Panimalar institute technology, Chennai. His area interest in Image processing and Network security, operating system .

A.Rengarajan received the Undergraduate Degree in Computer Science and Engineering from Madurai Kamaraj University, in 2000 and the Post Graduate degree in Computer Science and Engineering from Sathyabama University, Chennai in 2005 and Ph.D in Computer Science and Engineering at Bharath University, Chennai during 2011. He has more than 15 publications in National Conferences and international journal proceedings. He has more than 12 years of teaching experience. His areas of interest include Network security, Mobile computing, Data Structures, DBMS, Distributed systems and Operating systems.