International Journal on Applications in Information and Communication Engineering
Volume 2: Issue 1: January 2016, pp 14-19. www.aetsjournal.com ISSN (Online) : 2394-6237

Incursion Coverage over Information Realize Dataset Using Neural Network

M.Karthikeyan, Dr.N.Satish,

Abstract— A neural network may be a system composed of the many easy process parts operative in parallel whose operate is set by network structure, affiliation strengths, and therefore the process performed at computing parts or nodes. The weakness of neural network primarily based approaches is that if the dimension of the computer file is extremely large then it's troublesome for it to interpret the connection between inputs and outputs. The project proposes a synthetic Neural Network approach for intrusion detection. A feed forward neural networks trained by data gain formula was developed to classify the intrusion employing a profile knowledge set(KDD cup-99).Our approach will effectively find network anomalies and bring home the bacon high detection chance and low false alarms rate. Icon jointly illustrate that neural networks will expeditiously find this activity. I take a look at our systems against denial of service attacks, distributed denial of service attacks, and port scans. During this work, I explore network primarily based intrusion detection victimisation classifying, self-organizing maps for knowledge bunch and MLP neural networks for detection. The project goes to be a promising intrusion detection system in terms of detection accuracy, machine expense and implementation for period intrusion detection. The project detects the intrusion supported the information primarily based dataset victimisation neural network. This application is developed for many on-line commerce firms and e-commerce sites were clean up briefly as a result of major packet flood attacks, conjointly called Denial-of-Service (DoS) attacks, inflicting these firms to lose revenue, client satisfaction, and trust.

Keywords— Intrusion Detection, Feed Forward NeuralNetwork, KDD Cup'99 data, MLP.

I. INTRODUCTION

Computer attacks are now commonplace. By connecting your computer to the Internet, you increase the risk of having someone break in, install malicious programs and tools on it, and possibly use it to attack other machines on the Internet by controlling it remotely.

Several major banks have been subject to attacks, in which attackers gained access into customers' accounts and viewed detailed information about the activities on these accounts. In some instances the attackers stole credit card information to blackmail e-commerce companies by threatening to sell this information to unauthorized entities. Several online trading companies and e-commerce sites were shut down temporarily due to major packet flood attacks, also known as Denial-of-Service (DoS) attacks, causing these companies to lose revenue, customer satisfaction, and trust. A major software development company discovered that attackers had broken into its network and stolen the source code for future releases of its popular products. Just recently, the source code of the future flagship product belonging to a major software development company was stolen and made publicly available on the Internet.In order to combat this growing trend of computer attacks, both academic and industry groups have been developing systems to monitor networks and systems and raise alarms of suspicious activities

.*Overview of Neural Network*

A neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.Characteristics of Neural Nets:The good news: They exhibit some brain-like behaviors that is difficult to program directly likeLearning, Association, Categorization, Generalization, Feature extraction, Optimization, Noise immunity.The bad news is that neural nets are black boxes difficult to train in some cases.

Advantages

- A neural network can perform tasks that a linear program cannot.
- When an element of the neural network fails, it can continue without any problem by their parallel nature.
- A neural network learns and does not need to be reprogrammed.
- It can be implemented in any application.
- It can be implemented without any problem.

Neural Network Architectures

- Multi-Layer Perceptron (Back-Prop Nets) 1974-85
- Neocognitron 1978-84
- Adaptive Resonance Theory (ART) 1976-86
- Self-Organizing Map 1982
- Hopfield 1982
- Bi-directional Associative Memory 1985
- Boltzmann/Cauchy Machine 1985
- Counter propagation 1986
- Probabilistic Neural Network 1988
- General Regression Neural Network 1991
- Support Vector Machine 1995

The error information is fed back to the system which makes all adjustments to their parameters in a systematic fashion

Mr.M.Karthikeyan, Assistant Professor, Department of Information Technology,, Mahendra College of Engineering, Salem, (Email: mkarthick04@gmail.com0

Dr.N.Satish,, Professor/Head of Information Technology,, Mahendra College of Engineering, Salem. (Email: satishmail12@gmail.com)

(commonly known as the learning rule). This process is repeated until the desired output is acceptable. It is important to notice that the performance hinges heavily on the data. Hence, this is why this data should pre-process with third party algorithms such as DSP algorithms.

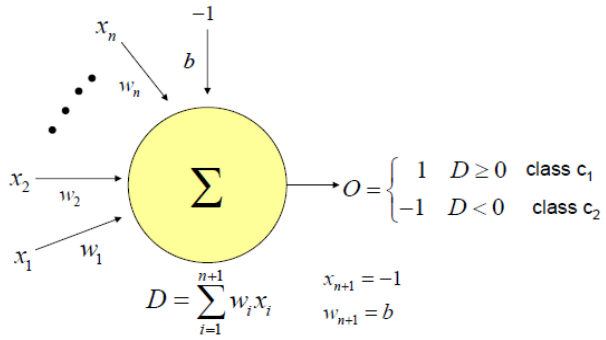

Fig.1 Single Neuron Model

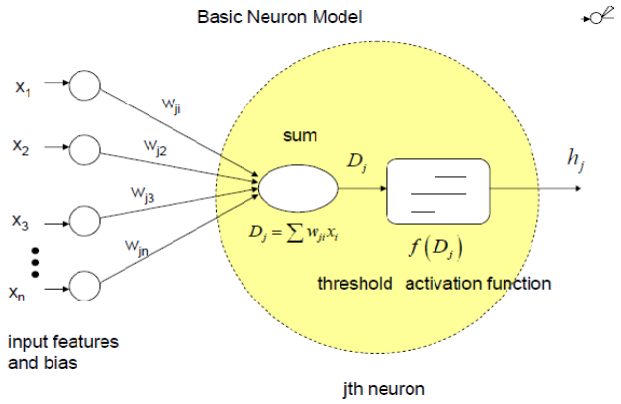

Fig.2Basic Neuron Model

In neural network design, the engineer or designer chooses the network topology, the trigger function or performance function, learning rule and the criteria for stopping the training phase. So, it is pretty difficult determining the size and parameters of the network as there is no rule or formula to do it. The best we can do for having success with our design is playing with it. The problem with this method is when the system does not work properly it is hard to refine the solution. Despite this issue, neural networks based solution is very efficient in terms of development, time and resources. By experience, I can tell that artificial neural networks provide real solutions that are difficult to match with other technologies.

Fifteen years ago, Denker said: "artificial neural networks are the second best way to implement a solution" this motivated by their simplicity, design and universality. Nowadays, neural network technologies are emerging as the technology choice for many applications, such as patter recognition, prediction, system identification and control.

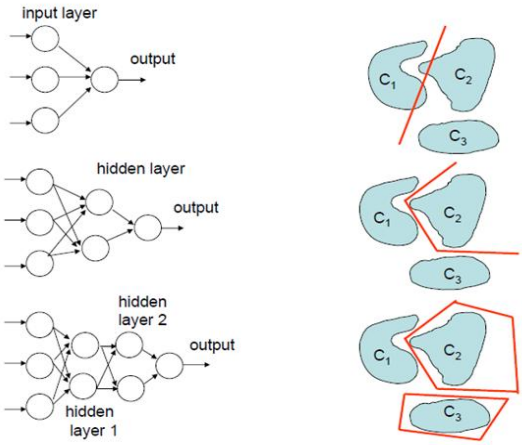

Fig.3Different Layers in Neural Network

Now that they have seen the biological neurons, we can talk about the artificial models of them. First, the synapses that we talked about before are modeled as weights. For an artificial neuron, these weights are in fact numbers. From before, we acknowledged that the neuron can have only one output: either it fires – either not. But, also, the neuron can have one or more inputs, so from this is clear that they actually are talking about a mathematical function, which is called activation function. Also, there is another very important attribute in every neuron – the threshold

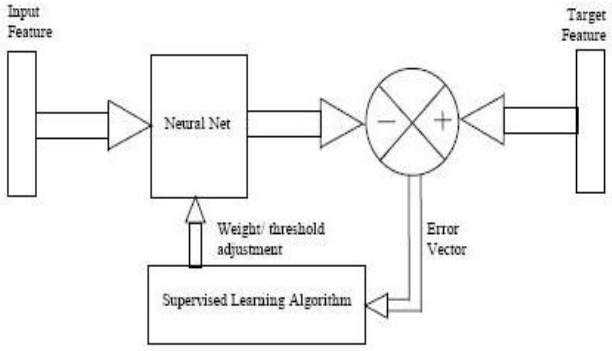

Fig.4Overall Process

Supervised learning or Associative learning in which the network is trained by providing it with input and matching output patterns. These input-output pairs can be provided by an external teacher, or by the system which contains the neural network.

Unsupervised learning or Self-organization in which an (output) unit is trained to respond to clusters of pattern within the input. In this paradigm the system is supposed to discover statistically salient features of the input population. Unlike the supervised learning paradigm, there is no a priori set of categories into which the patterns are to be classified; rather the system must develop its own representation of the input stimuli.

Reinforcement Learning This type of learning may be considered as an intermediate form of the above two types of

learning. Here the learning machine does some action on the environment and gets a feedback response from the environment. The learning system grades its action good (rewarding) or bad (punishable) based on the environmental response and accordingly adjusts its parameters. Generally, parameter adjustment is continued until an equilibrium state occurs, following which there will be no more changes in its parameters. The self-organizing neural learning may be categorized under this type of learning.

Application of the Neural Networks in Chemo Informatics

First, they will make this "chemo informatics" term clear. Chemo informatics (also known as chemo informatics and chemical informatics) is the use of computer and informational techniques, applied to a range of problems in the field of chemistry.

The number of applications of the neural networks in chemo informatics Increases every day. In general, different types of problems require different neural network architectures and different learning strategies. An interesting fact though, is that about 90% of the problems described until now in the chemical literature have used one-hidden-layer neural networks and the back-propagation learning strategy.

I. PROBLEM STATEMENT

The Intrusion Detection in Computer Networks is based on using Artificial Neural Network (ANN) for detecting the Normal and Abnormal conditions of the given parameters, which leads to various attacks. The neural network approach for this purpose has two phases; training and testing.

During the training phase, neural network is trained to capture the underlying relationship between the chosen inputs and outputs. After training, the networks are tested with a test data set, which was not used for training. Once the networks are trained and tested, they are ready for detecting the intrusions at different operating conditions.

1. Data Collection

DARPA Intrusion Detection Evaluation– MIT Lincoln Laboratory. There are two ways to build IDS, one is to create our own simulation network, and collect relevant data and the other one is by using previously collected datasets. Issues like privacy, security, and completeness greatly restrict people from generating data. The beauty of using previously collected datasets is that the results can be compared with others in the literature. Not many data sets have being collected that could build IDS systems. Some of the popularly used IDS datasets are DARPA 1998 data set, DARPA 1999data set and KDD Cup 1999 data set which are available in the MIT Lincoln Labs.

2. Data Preprocessing

Before training the neural network [7], the dataset should be preprocessed to remove the redundancy present in the data and the non-numerical attributes should be represented in numerical form suitably.

3. Data Normalization

During training of the neural network [2], higher valued input variables may tend to suppress the influence of smaller ones. Also, if the raw data is directly applied to the network, there is a risk of the simulated neurons reaching the saturated conditions. If the neurons get saturated, then the changes in the input value will produce a very small change or no change in the output value. This affects the network training to a great extent. To minimize the effects of magnitudes among inputs as well as to prevent saturation of the neuron activation function, the input data is normalized before being presented to the neural network. One way to normalize the data x is by using the expression: where, xn is the normalized value and xmin and xmax are the minimum and maximum values of the data.

4. Selection of Network Structure

To make a Neural Network to perform some specific task [8], one must choose number of input neurons, output neurons, hidden neurons and how the neurons are connected to one another. For the best network performance, an optimal number of hidden-units must be properly determined using the trial and error Procedure. The hidden layer neurons have tangent hyperbolic function as the activation function and the output have linear activation function.

5. Network Training and Testing

Once the appropriate structures of the network are selected, the ANN model [2] is trained to capture the underlying relationship between the input and output using the training data. In this work, Back propagation algorithm is used to train the network, which propagates the error from the output layer to the hidden layer to update the weight matrix. After training, the Networks are tested with the test data set to assess the generalization capability of the developed network.

6. Review of Artificial Neural Network

Artificial Neural Networks can be viewed as a parallel and distributed processing system which consists of a huge number of simple and massively. The MLP architecture [4] is the most popular paradigm of artificial neural networks in use today. Figure 1 shows a standard multilayer feed forward network with three layers. The neural network architecture in this class shares a common feature that all neurons in a layer are connected to all neurons in adjacent layers through unidirectional branches. That is, the branches and links can only broadcast 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 back propagation algorithm. Training the network with back propagation algorithm results in a non-linear mapping between the input and output variables. Thus, given the input/output pairs, the network can have its weights adjusted by the back propagation algorithm to capture the non-linear relationship. After training, the networks with fixed weights can provide the output for the given input. The standard back propagation algorithm for training the network is based on the minimization of an energy function representing the instantaneous error. In other words, we desire to minimize a function defined as where dq represents the desired network output for the qth input pattern and yq is the actual output of the neural network. Each weight is changed according to the rule: where, k is a constant of proportionality, E is the error function and wij represents the weights of the connection between neuron j and neuron i.

II.PPROACH

In this Concept, an intrusion detection system called denial of service Intelligent Detection (DoSID) is developed. The type of Neural Network used implement DoSID is feed forward which uses the back propagation learning algorithm. The data used in training and testing is the data collected by Lincoln labs at MIT for an intrusion detection system evaluation sponsored by the U.S defense Advanced Research Project Agency (DARPA).Results show that the best detection rate for new attacks is 68%.Also it has been shown in the final experiment that the false positive of the system has been reduced considerably.

The proposed methodology for Intrusion Detection in Computer Networks is based on using Artificial Neural Network (ANN) for detecting the Normal and Abnormal conditions of the given parameters, which leads to various attacks. The neural network approach for this purpose has two phases; training and testing. During the training phase, neural network is trained to capture the underlying relationship between the chosen inputs and outputs. After training, the networks are tested with a test data set, which was not used for training. Once the networks are trained and tested, they are ready for detecting the intrusions at different operating conditions. The proposed Approach is well in detecting R2L and U2R attack.

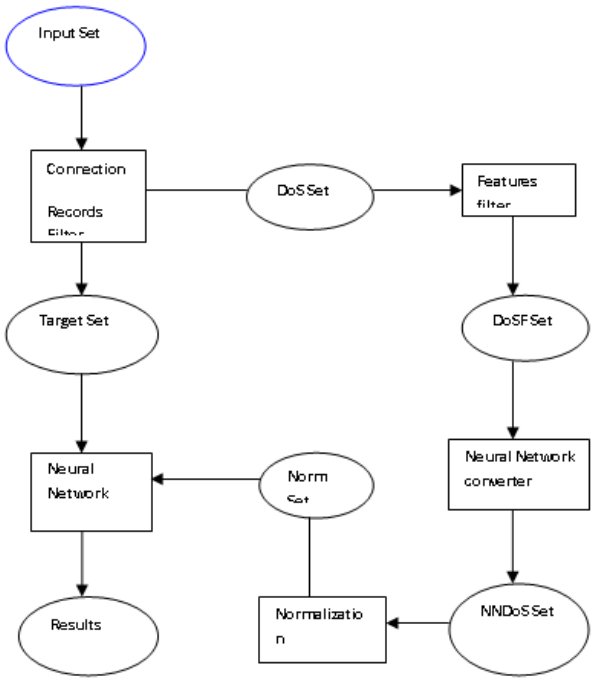

Fig.5 System Architecture

III. ARCHITECTURAL DESIGN

The major part of the project development sector considers and fully survey all the required needs for developing the project. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations. Generally algorithms shows a result for exploring a single thing that is either be a performance, or speed, or accuracy, and so on. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them.

IV. METHODOLOGY

Following are the most frequently used project management methodologies in the project management practice:

1. Feature Extraction
2. Training and Testing Dataset
3. Performance Measurement

1) Feature Extraction

Feature selection is done based on the contributionthe input variables make to the constructionof the decision tree. Feature importance isdetermined by the role of each input variableeither as a main splitter or as a surrogate. Surrogatesplitters are defined as back-up rules thatclosely mimic the action of primary splitting rules.Suppose that, in a given model, the algorithmsplits data according to variable 'protocol type'and if a value for 'protocol type' is not available,the algorithm might substitute 'service' as a goodsurrogate. Variable importance, for a particular variable is the sum across all nodes in the tree ofthe improvement scores that the predictor haswhen it acts as a primary or surrogate (but notcompetitor) splitter.

The data for our experiments were prepared by the 1999 DARPA intrusion detection evaluationprogram by MIT Lincoln Laboratory. The data setcontains 24 attack types that could be classifiedinto four main categories namely Denial of Service (DOS), Remote to User (R2L), User to Root (U2R) and Probing. The original data contain 744 MB datawith 4,940,000 records. The data set has 41attributes for each connection record plus one.

To Calculate Information Gain:

Class Entropy:

$$MI(S1,S2\ldots Sn) = -\sum_{i=1} Si/S \log(Si/S)$$

- S=total no of records
- Si=no of records for ith class
- M=total no of class

Feature Entropy:

$$vE(f) = \sum_{s=1} (Sis+\ldots+Sms/S)*I(Sis\ldots Sms)$$

- f1,f2,…fv =unique value of feature
- s1,s2,…sv=unique value of feature count.

2) Training and Testing Dataset

$$G(f) = I(S1\ldots Sm)-E(f)$$

ANeural Network is a structure which is composed of a number of simple elements or node called neurons
These elements are always operating in parallel. The function of the neural Network is determined largely by the connection between the neurons. These neurons are connected by inks and each link is adjusted by values called weighs. The process of updating the weights is called learning.

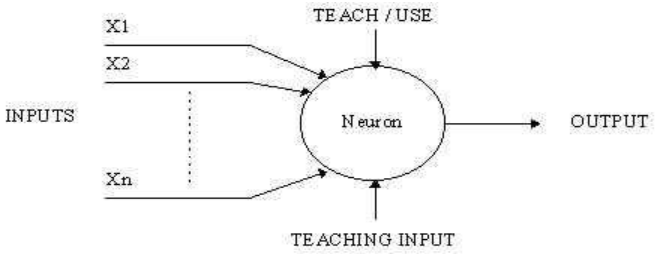

Fig.6 Simple Neuron

Feed-forward networks usually consist of two to three layers in which the neurons are logically arranged. The last layer is the output layer and there are usually one or more hidden layers before the output layer. The DoSID Neural Network as shown in Fig is composed of two layers (the hidden and the output layer), a variable number of neurons in the hidden layer and here is one neuron in the output layer. Each output vector element value is in the range [-1,1]. The transfer functions of neurons on both layers are "tan-sigmoid" function. This function takes the input, which may have any value between plus and minus infinity, and squashes the output into the range [-1,1]. The input vector contains 31 elements. These elements are the result of converting the 18 features in the DARPA dataset to Neural Network format.
The most common and widely used learning algorithm for multilayer feed forward Neural Networks is the back propagation algorithm.
It is based on the Delta Rule that basically states that if the difference (delta error) between the user's desired output (target) and the network's actual output is to be minimized, the weights must be continually modified. The error in the output layer has been adjusted, and therefore it can be used to change the input connection weights so that the desired output may be achieved.
This is why feed-forward networks are also often called "back propagation feed-forward networks".
The input connection weights are adjusted in such a way that the delta error will be minimized. This process is repeated several times (Iterations). The training stops if: the number of iterations exceeds a certain number of iterations, the training performance function drops below certain threshold of MSE, or the training time is longer than certain threshold of seconds. The mean squared error (MSE) is computed by summing the squared differences between the target and the network's actual output, and then dividing the sum by the number of components (input vector elements) that went into the sum.
Data sets were preprocessed by extracting the IP packet header information to create feature vectors. Theresulting feature vectors were used to calculate the principal components and other statistics. The featurevector chosen has the following format

SIPx	SPort	DIPx	DPort	Prot	PLen

Where

- SIP*x*= Source IP address nibble, where *x* = [1-4]. Four nibbles constitute the full source IP address
- SPort = Source Port number
- DIP*x*= Destination IP address nibble
- DPort = Destination Port number
- Prot = Protocol type: TCP, UDP or ICMP
- PLen = Packet length in bytes

This format represents the IP packet header information. Each feature vector has 12 componentscorresponding to dimension *d* in the PCA equation for original data vector *x*. The IP source and destinationaddresses are broken down to their network and host addresses to enable the analysis of all types ofnetwork addresses.
Seven data sets were created, each containing 300 feature vectors as described above. Four data sets represented the four different attack type's one each of shown in Table 5.1 The three remaining data setsrepresent different portions of normal network traffic across different weeks of the DARPA Data Sets. Thisallows for variations of normal traffic to be accounted for in the experiment.One of the motives in creating small data sets (i.e. 300 feature vectors each) for representing the featurevectors is to study the effectiveness of this method for real-time applications. Real-time processing ofnetwork traffic mandates the creation of small sized databases that are dynamically created from real-timetraffic presented at the network interface. For real-time applications, each 300 packets arriving at thenetwork interface are collected and preprocessed to extract the IP header information, producing 300feature vectors that are analyzed using PCA as a single small data set. Then the process is repeateddynamically as more packets arrive at the network interface, producing a dynamic method for analyzing thetraffic. Since DARPA data is only available statically, seven small data sets were created to mimic the case of dynamic real-time operation.

With each packet header being represented by a 12 dimensional feature vector, it is difficult to view thishigh-dimensional vector graphically and be able to extract the relationships between itsvarious features. Itis equally difficult to extract the relationship between the many vectors in a set. Therefore, the goal of usingPCA is to reduce the dimensionality of the feature vector by extracting the PCs and using the first andsecond components to represent most of the variance in the data. This graphical representation would enable bettervisualization of the summary of the relationships in the data set. This visualization is achieved using Biplots.
PCA was performed on all data sets where each feature vector would be represented by its 12 components.The following statistics was generated for each data set:

- Standard Deviation for each component
- Proportion of variance for each component
- Cumulative proportion of variance across all components
- Loading value of each feature on all individual components
- A Bi-plot representing the loading of the different features on the first and second components.

3) Performance Measurement

Table 1 shows the execution time of the PCA algorithm including the display time for the Bi-plots graphics. Three times are shown: User time indicates the time (in seconds) consumed for the user process, system time indicates the time consumed by the operating system and the elapsed time indicates the total time consumed by the overall operation. The difference between user time and system time is that user time is the CPU time used while executing instructions in the user space of the calling process, while systemtime is the CPU time used by the system on behalf of the calling process. With a total elapsed time of 1.59 seconds used in processing 1200 packets, the method can process roughly 1000 packets per second. These performance numbers may not be adequate in handling real-time network traffic. Utilizing a more powerful computer system with advanced graphics hardware for executing the algorithm and displaying the Bi-plot scan further enhance the performance of the method.

IDS Performance measure

	POSITIVE	NEGATIVE
TRUE	Normal without threat	Normal but IDS detect wrong
FALSE	Find attack and discard	Attack packet but IDS execute

Table.1 IDS Performance Measure

V. CONCLUSION AND FUTURE WORK

The input data used in this work contains 9 input features which may possible to have redundant data and false correlations which hinder the process of detecting intrusions. To make the Neural Network applicable to very large data set, some dimensionality reduction is mandatory. Hence as an enhancement to this proposed work, future research will be directed towards the dimensionality reduction techniques which remove the unwanted input futures and select only the optimal feature for training the neural network there by increases the detection rate.

II. REFERENCES

[1] SrilathaChebrolu, Ajith Abraham and Johnson P. Thomas. Feature deduction and ensemble design of intrusion detection system. Computers &Security, Volume 24, Issue 4, June 2005, Pages 295-307.

[2] Analoui.M, A.Mizaei, and P.Kabiri, "Intrusion detection using multivariate analysis of variance algorithms," in Third International Conference on Systems, Signals & Devices SSD05, vol. 3, Sousse, Tunisia. Mar. 2005, IEEE.

[3] Zhong.A and C.F. Jia, "Study on the applications of hidden Markov models to computer intrusion detection" in Proceedings of the Fifth World Congress on Intelligent Control and Automation WCICA, vol.5, pp. 4352-4356. IEEE, June 2004.

[4] M.Ramadas, S.Ostermann, and B.Tjaden, "Detecting anomalous network traffic with self organizing maps", in Recent Advances in Intrusion Detection, 6thInternational Symposium, RAID 2003, pages 36-54, 2003.

[5] Lee, W., Xiang, D.: Information-Theoretic Measures for Anomaly Detection. Proceedings of the IEEE Symposium on Security and Privacy, IEEE Computer Society Press, Oakland, CA (2001) 130-143

[6] Bonoficio J.M, "Neural Networks Applied in Intrusion Detection Systems", Neural Networks Proceedings, IEEE World Congress on Computational Intelligence, vol. 1, pages:205-210, 1998.