International Journal of Soft Computing 9 (3): 117-121, 2014
ISSN: 1816-9503

A Self Learning Algorithm for Anomaly Based Intrusion Detection System using Genetic Neural Network

[1]M. Ravichandran and [2]C.S. Ravichandran
[1]Department of ECE, Saveetha School of Engineering, Saveetha University, Chennai, India
[2]Department of EEE, Sri Ramakrishna Engineering College, Coimbatore, India

Abstract: An Anomaly based Intrusion Detection System is a one which monitors the system or network traffic looking for anomalous behaviour rather than matching the user behaviour pattern alone. Hence, the Anomaly Based Intrusion Detection algorithms have the capability to extend their detection mechanisms to detect unknown attacks. In this research, a Self Learning algorithm for anomaly based Intrusion Detection Model which is based on genetic neural network is proposed. The genetic neural network combines the good global searching ability of Genetic algorithm with the accurate local searching feature of back propagation neural networks. Here, it is used to optimize the initial weights of the neural network. The scope of the algorithm in this proposed research remains in identifying the malicious packet.

Key words: Intrusion Detection System, neural network, Genetic algorithm, genetic neural network, network traffic

INTRODUCTION

An Intrusion Detection System (IDS) monitors the system or network traffic for suspicious activity and alerts the system or network administrator (Das *et al*., 2008). In certain cases, the IDS also responds to anomalous traffic by blocking the source IP address from accessing the network. An IDS exists in a variety of flavors and aim at detecting the suspicious traffic in different ways. They are Network based IDS (NIDS) and Host based (HIDS) Intrusion Detection Systems. There are IDS that detect based on specific signatures of known threats very much similar to the way antivirus software detects and protects against malwares and there are IDS that detect based on comparing traffic patterns against a baseline and looking for anomalies. The former is called as signature based IDS and the latter is known as anomaly based IDS.

TYPES OF INTRUSION DETECTION SYSTEM

Network Intrusion Detection System: Network Intrusion Detection Systems are placed at a premeditated point or points within the network to monitor traffic to and from all devices on the network. Ideally the best option is to scan all incoming and outgoing traffic, however on doing so it might create a major bottleneck that would impair the overall speed of the network.

Host Intrusion Detection System: Host Intrusion Detection Systems are run on individual machines or devices on the network system. A HIDS primarily monitors the incoming and outgoing packets from the device and will alert the user or administrator of leery activity detected in the host machine.

Signature based: A signature based IDS will supervise packets on the network and compare them against a repository of signatures or attributes from known malicious threats. This is similar to the way by which most antivirus software detects malware. The problem is that there will be a time lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to your IDS. During this time lag your IDS would not be able to detect the new threat.

Anomaly based: An anomaly based IDS will monitor network traffic and compare it against an already available or pre-established baseline. The baseline will identify what is normal for that network, i.e. what sort of bandwidth and what protocols are generally used through which ports the devices generally connect to each other and alert the administrator or user when anomalous traffic is detected which is significantly different than the baseline.

EXISTING MACHINE LEARNING ALGORITHMS

Machine learning techniques are based on establishing an explicit or implicit model that enables the patterns analyzed to be categorized. A singular characteristic of these schemes is the need for labelled

Corresponding Author: M. Ravichandran, Department of ECE, Saveetha School of Engineering, Saveetha University, Chennai, India

data to train the behavioural model, a procedure that places severe demands on resources. In many cases, the applicability of machine learning principles coincides with that for the statistical techniques, although the former is focused on building a model that improves its performance on the basis of previous results. Hence, a machine learning Anomaly based Network IDS (A-NIDS) has the ability to change its execution strategy as it acquires new information.

Although, this feature could make it desirable to use such schemes for all situations, the major drawback is that they are resource expensive in nature. Several machine learning-based schemes have been applied to A-NIDS. Some of the most important are cited and their main advantages and drawbacks are identified.

Neural networks: With the aim of simulating the operation of the human brain (featuring the existence of neurons and of synapses among them), neural networks have been adopted in the area of anomaly intrusion detection, chiefly because of their inherent flexibility and adaptability to environmental changes. This detection approach has been employed to create user profiles, to predict the next command from a sequence of previous ones, to identify the intrusive behaviour of traffic patterns, etc. However, a common characteristic in the proposed variants from recurrent neural networks to self-organizing map is that they do not provide a descriptive model that explains why a particular detection decision has been taken.

In more practical terms, neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. The ability to learn and adapt to uncertainties of ANN are just suitable to solve the intrusion detection problem (Jiang and Ruan, 2009).

Fuzzy logic techniques: Fuzzy logic is derived from fuzzy set theory under which reasoning is approximate rather than precisely deduced from classical predicate logic. Fuzzy techniques are hence used in the area of anomaly detection mainly because the features to be considered can be seen as fuzzy variables. This type of processing scheme considers an observation as normal if it lies within a given interval. Although, fuzzy logic has proven to be very effective, especially against probes and port scans, its main disadvantage is the high resource consumption. On the other hand, it should also be noticed that fuzzy logic is controversial in certain cases and it has been rejected by some engineers and by most statisticians who firmly believe that probability is the only rigorous mathematical description of uncertainty.

Genetic algorithms: Genetic algorithm is a family of computational models based on principles of evolution and natural selection (Li, 2004). GA's are categorized as global search heuristics and are a unique class of evolutionary algorithms (also known as evolutionary computation) that use techniques inspired by evolutionary biology like inheritance, mutation, selection and finally recombination. Thus, Genetic algorithms form another type of machine learning-based scheme, capable of arriving at classification rules and/or selecting appropriate features or optimal parameters for the detection process. The major merits of this subtype of machine learning ANIDS is the use of a flexible and robust global search method that converges to a solution from different multiple directions while no prior knowledge about the system behaviour is assumed. Its main demerit is the high resource consumption involved.

LITERATURE REVIEW

Security of an information system is so vital in today's era where computers are interconnected via internet. As no system can be fully secure (Vesely and Brechlerova, 2004), the timely and accurate detection of intrusions seems to be of utmost necessity. For this purpose, Intrusion Detection Systems were designed. There are two fundamental models of IDS: misuse IDS and anomaly IDS. Artificial neural networks offer the potential to resolve these problems. As far as anomaly based systems are concerned, it is so difficult to build them, since it is very hard to define the normal and abnormal behaviour of a system. Also, for constructing the anomaly system, neural networks can be used since they can learn to discriminate the normal and anomalous behaviour of a system from examples. Therefore, they offer a guaranteed technique for Building Anomaly Systems.

With the ability of strong self-learning and quicker convergence of high speed and precise Genetic algorithm neural network, the network intrusion detection technique can detect various intrusion behaviours rapidly and effectively by learning the typical intrusion characteristic information.

PROPOSED RESEARCH

Genetic neural network: An Intrusion Detection algorithm which combines the features of Backpropagation Neural Network (BPNN) and Genetic algorithm is proposed. The traditional BpNN algorithm is used in many decision making applications for is accuracy and fast self learning ability and the Genetic algorithm are used in application where the solution space is large and accurate global searching is needed. The proposed methodology using Genetic Neural Network (GNN) combines the good global searching ability of Genetic algorithm with the accurate local searching feature of back propagation neural networks to optimize the initial weight of neural networks.

Adaptive Genetic algorithm: Adaptive Genetic Algorithm (AGA) is a kind of Genetic algorithm that has scale reproduction and self-adaptive crossover and mutation operations. In the process of searching for the optimum parameter, AGA changes the crossover probability and mutation probability adaptively according to the different condition of individuals in order to keep the diversity of colony and prevent the premature convergence, further it can enhance the calculating speed and precision of the algorithm (Tian and Gao, 2009):

$$\text{Pc} = \begin{cases} \dfrac{\text{k1}(f_{\max} - f')}{f_{\max} - f_{\text{avg}}} & \text{if } f' > f_{\text{avg}} \\ \text{k3} & \text{if } f' < f_{\text{avg}} \end{cases} \tag{1}$$

$$\text{Pm} = \begin{cases} \dfrac{\text{k2}(f_{\max} - f)}{f_{\max} - f_{\text{avg}}} & \text{if } f > f_{\text{avg}} \\ \text{k4} & \text{if } f < f_{\text{avg}} \end{cases} \tag{2}$$

Where:

Pc = Probability of crossover
Pm = Probability of mutation
$f_{\max}$ = The biggest fitness of colony
f_{avg} = The average fitness of colony
f' = The bigger fitness of two strings used for exchange in crossover
f = The fitness of the individual to mutate
k1 to k4 = Parameters that control the adaptive nature of Pc and Pm

Generally, k1 = k3 = 1, k2 = k4 = 0.5. At practical application, the value of Pc is often in range 0.5-1.0 and the Pm in range (0.005-0.05).

Structure of the neural network: Structure of the neural network used in the proposed system consist of one input layer with 11 input neurons corresponding to the eleven features taken into consideration from the 41 input dataset of the KDD CUP'99 and an output layer with one output neuron. The output neuron decides whether an intrusion is detected or not. Only one hidden layer with six hidden neurons is used.

Careful attention is paid in selecting the number of nodes in the hidden layer as it determines the non-linear mapping function, fault tolerance and also the time required for learning. Very few nodes in the hidden layer leads to poor fault tolerance leading to frequent false alarms and too many nodes may lead to increased learning time. Hence, a tradeoff is made between the fault tolerance and learning time. To achieve proper tradeoff and balance, it is presumed that the number of nodes in the hidden layer could be the mean of the number of nodes in input and output layers.

Activation function: To introduce non-linearity into the network, activation functions for the hidden units are required. With no nonlinearity, hidden units cannot make nets more powerful than just plain perceptrons. The main reason is that a linear function of a linear function is again a linear function. However, it is the non-linearity (i.e., the capability to represent non-linear functions) that makes multilayer networks extremely powerful. Almost any nonlinear function does the job, however except for the polynomials. For the backpropagation learning, the activation function must be a differentiable one and it is most helpful if the function is bounded, the sigmoidal functions like logistic and tanh and the Gaussian function are the most preferred choices. Both positive and negative values are produced by functions like tanh or arctan and these are likely to produce much quicker training than functions like logistic that produce only positive values because of good numerical conditioning.

Sigmoid activation functions are usually preferable to threshold activation functions for hidden units. Networks with threshold units are much difficult to train since the error function is stepwise constant hence the gradient either do not exist or it is zero, making it impossible to use backpropagation or more efficient gradient-based training methods. Even for the training methods which does not use gradientssuch as simulated annealing and genetic algorithms, sigmoid units are much easier to train than threshold units. With sigmoid units, a minor variation in the weights will yield a change in the outputs which makes it possible to suggest whether that change in the weights is good or bad whereas with threshold units, a minor deviation in the weights will most often produce no change in the outputs.

Genetic Neural Network algorithm: Figure 1 shows the training module using Genetic Neural Network algorithm.

Description of the algorithm: The flow diagram of the process involved in the identification of intrusion is depicted in the Fig. 2.

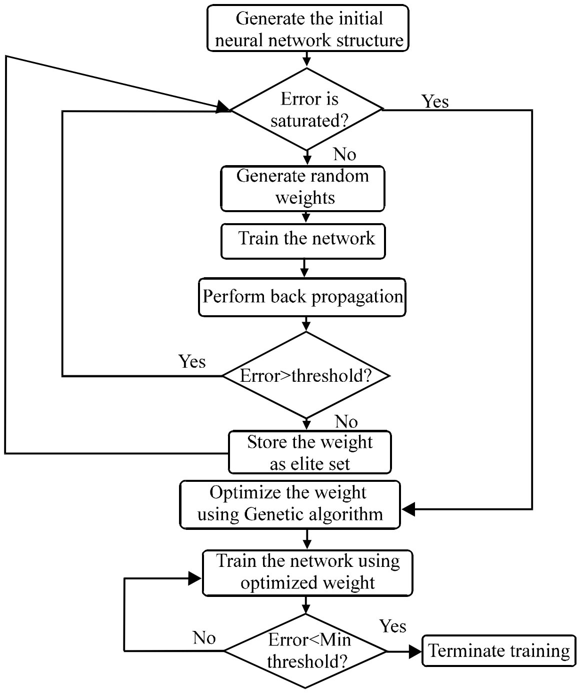
Generate the initial neural network structure
Error is saturated?
Yes
No
Generate random weights
Train the network
Perform back propagation
Yes
Error>threshold?
No
Store the weight as elite set
Optimize the weight using Genetic algorithm
Train the network using optimized weight
No
Yes
Error<Min threshold?
Terminate training

Fig. 1: Training module using genetic neural network

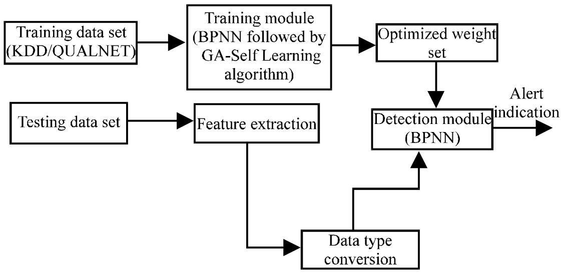

Fig. 2: Flow diagram of the Intrusion Detection System

Step 1: The attributes (e.g., for identifying class 3 based DOS attack, 11 attributes chosen from 41 inputs of the KDD CUP'99 dataset are duration, src_bytes, dst_bytes, count, srv_count, serror_rate, srv_serror_rate, dst_host_count, dst_host_same_src_port_rate, dst_host_serror_rate, dst_host_srv_serror_rate) for the various attacks are extracted from the trace file (KDD/QUALNET) using the Feature Extraction Method. KDD/QUALNET trace file is used as the training data for the proposed IDS. QUALNET Software can be used to simulate real time network scenarios include malicious nodes and obtain simulated traces which can be used for training the detection system.

Step 2: The extracted attribute values mentioned in step 1 are mapped/scaled to form input neuronal data. The initial weights for the extracted attributes are randomly chosen. The structure of the BPNN will consist of 11 input neurons (11 extracted neuronal data), 6 hidden neurons and an output neuron for classifying the packet as attack/no attack.

Step 3: The actual output from neural network output layer is compared with the target output available from KDD/QUALNET. Error value is computed and is compared with the desired set threshold. If error is less than the desired set threshold, the iteration is continued and the weights used for this iteration are stored as elite weights. The process is continued till saturation or convergence is reached. The elite weights stored are used to form the input chromosomes in GA. If error is greater than the desired set threshold, iteration is continued and the weights are not stored. After every iteration, the error value is back propagated which results in the readjustment of weights.

Step 4: The best set of weights to detect the specific threat is identified with the help of Genetic algorithm as follows: each input neuron (11 input neurons) is connected to all hidden neurons (6 hidden neurons) thereby forming 66 weighted neuronal links. Each neuron would correspond to a unique gene in GA and each gene would comprise of 6 nucleotides which are the weighted links between one input neuron and 6 hidden neurons. Altogether, a parent chromosome would consist of 11 such genes. Two parent chromosomes are subject to genetic operations (roulette wheel based selection crossover, reproduction and mutation). GA optimized weights are given as training weights for BPNN.

Step 5: Error value is computed and the weight for the least error value is stored. The least error value corresponds to the best optimal weight that is used for detection of real time data (for specific threat/attack for all packets). This completes the training part of the Neural Network System with the aid of Genetic algorithm.

Step 6: For detection using BPNN, the real time test data is provided as the input. The input value is multiplied with the already trained (by BPNN and GA) and optimized weight (by GA) for the specific attack. If the output obtained from BPNN is less than the threshold set for the specific attack, then the packet is not malicious. Otherwise, the packet is identified as a malicious packet.

CONCLUSION

In this study, a Self Learning algorithm for anomaly based Intrusion Detection System which combines the merits of back propagation Neural Network and Genetic algorithm has been proposed.

REFERENCES

Das, A., D. Nguyen, J. Zambreno, G. Memik and A. Choudhary, 2008. An FPGA-based network intrusion detection architecture. IEEE Trans. Inform. Forens. Secur., 3: 118-132.

Jiang, H. and J. Ruan, 2009. The application of genetic neural network in network intrusion detection. J. Comput., 4: 1223-1230.

Li, W., 2004. Using genetic algorithm for network intrusion detection. Proceedings of the United States Department of Energy Cyber Security Group 2004 Training Conference, May 24-27, 2004, Kansas City, Kansas, USA., pp: 1-8.

Tian, J. and M. Gao, 2009. Network intrusion detection method based on high speed and precise genetic algorithm neural network. Proceedings of ACM International Conference on Networks Security, Wireless Communications and Trusted Computing, 2009, Volume 2, April 25-26, 2009, Wuhan, Hubei, pp: 619-622.

Vesely, A. and D. Brechlerova, 2004. Neural networks in intrusion detection systems. Agric. Econ. Czech, 50: 35-39.