International Journal of Soft Computing 7 (2): 44-49, 2012
ISSN: 1816-9503

Design and Analysis of ANN-Based Echo State Network Intrusion Detection in Computer Networks

[1]S. Saravanakumar, [1]T.A. Mohanaprakash, [2]R. Dharani and [3]C. Jaya Kumar
[1]Department of CSE, [2]Department of IT, Panimalar Institute of Technology, Chennai, India
[3]Department of CSE, RMK Engineering College, Chennai, India

Abstract: The complexity of attacks on computer systems are increasing rapidly. The current network is complicated due to the high throughput and the multi-uniformity of actions. Intrusion detection is a process of monitoring the various computer networks and systems for violations of security and this can be automatically done with the help of an intrusion detection system. An Intrusion Detection System (IDS) is a critical component for secure information management. IDS plays a major role in detecting and disrupting various attacks before cooperating with the software. This study presents the investigations carried out on different neural network structures using a number of algorithms for intrusion detection. Also, this study proposed an Echo State Network (ESN) structures for intrusion detection. The proposed algorithm has faster convergence and better performance in IDS. The objective of this study is to implement the ESN algorithm and compare with other neural network algorithms in a networked environment. The performances of different methods have been implemented and compared using the Knowledge Discovery and Datamining (KDD) dataset to experiment the performance of ESN in classifying the Local Area Network (LAN) intrusion packets.

Key words: ANN, denial of service, Echo State Network intrusion detection, malicious attacks, neural networks

INTRODUCTION

A computer system should provide confidentiality integrity and assurance against Denial of Service (DoS). Due to increased connectivity and the vast spectrum of financial possibilities that are opening up, more and more systems are subject to attack by intruders. Any system connected to internet cannot provide security without additional provision of intrusion detection elimination softwares (Chavan *et al*., 2004). Every organization of even small size is connected to internet. Due to functional requirements and cost factors, employees work from their home by connecting their systems with the main office. Employees exchange data in the form of revision, completion of the work assigned to them. Financial organization, automatic teller machines, landline telephones, cellular phones and wireless networks provide internet facilities. The equipment which rely upon main database stored in severs should not be damaged due to software threats in the form of intrusion. Military bases, nuclear research centers organization with top level information should not be damaged in the form of alteration, corruption of information by any unknown activities entertained through the internet facilities by any one.

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents which are violations of computer security policies. Intrusion detection can be used to guard a host computer or network against being a source or a victim of an attack. Intrusion detection system is a software that automates the intrusion detection process. IDS has become increasingly vital over the last decade as network information systems have grown into the daily life of most businesses, government agencies and private citizens (Li *et al*., 2004). Because of the increasing dependence in which companies and government agencies have their own computer networks and the importance of protecting these systems from attack is critical. A single intrusion of a computer network can result in the loss of unauthorized utilization or modification of large amounts of data and cause users to question the reliability of all of the information on the network.

IDSs can be categorized into three types namely, network-based intrusion detection, router-based intrusion detection and host-based intrusion detection (Bu *et al*., 2011). Network-based intrusion detection, operate at the gateway of a network and examines all the incoming packets. Router-based intrusion detection is installed on

Corresponding Author: S. Saravanakumar, Department of CSE, Panimalar Institute of Technology, Chennai, India

the routers to prevent intruders from entering into the network. Finally, the host-based intrusion detection receives the necessary audit data from the hosts operating system and analyzes the generated events to keep the local system secure. A centralized scheme is proposed to schedule authentication and intrusion detection which needed a centralized controller (Liu *et al*., 2009). This is more opted for a single system rather than a network with distributed systems with random mobility. There are numerous methods of responding to a network intrusion (Jaeger, 2002a, b) but they all require the accurate and timely identification of the attack. IDS detect DoS attacks either by using a priori knowledge of the types of known attacks or by recognizing deviations from normal system behaviors. DoS attacks aim at denying or degrading legitimate users access to a service or network resource or at bringing down the servers offering such services.

An intruder can get into the system through primary intrusion, system intrusion and remote intrusion (Kou *et al*., 2006). The IDS responses to set of actions when detects intrusions. Some of the responses are mentioned in which involves reporting results and findings to a pre-specified location while others are more active automated responses (Helman and Liepins, 1993). IDS can be viewed as the second layer of protection against unauthorized access to networked information systems because despite the best access control systems and the intruders are still able to enter computer networks (Zhang *et al*., 2007). IDS expand the security provided by the access control systems by using system administrators with a warning of the intrusion (Xue *et al*., 2006).

Various algorithms are used to model the attack signatures and normal behavior response patterns of the systems. There are three algorithms used to model the various attacks and are named as naive Bayes, Artificial Neural Network (ANN) and Decision Tree (DT) (Katar, 2006). The naive Bayes classifier is based on a Probabilistic Model. This model will assign the most likely class for a given instance. ANN Model is a pattern recognition technique. This technique has the capacity to adaptively model the user or the system behavior. DT Model is a machine learning technique. This model is used to organize the attack signatures into a tree structure. IDS will create two kinds of errors. One is False Positive (FP) and another is False Negative (FN). FNs mainly results in security breaches since, intrusions are not detected. Therefore, no alert is raised. The False Negative Rate (FNR) is used to measure the secure characteristics of the IDS. A low FNR means a low possibility that intrusion can occur without detection (Bu *et al*., 2011).

The importance of the present research is to analyze the potential benefits of ANN algorithms as intrusion detection software in a computer network connected with internet facility. When an ANN is properly explored for its complete implementation in intrusion detection software, most of the attacks can be detected. Some of the attacks are: Attempted break-ins, Masquerade attacks, Penetration of the security control system, Leakage, Denial of Service and Malicious use, etc.

ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANN) has computing elements that are based on the structure and function of the biological neurons. The new algorithms are faster and give better performance. ANN consists of interconnected processing units. The general model of processing unit consists of summing part followed by an output part. The summing part receives n input values and weight values and performs a weighted sum. The weighted sum is called the activation value. The sign of the weight for each input determines whether the input is excitatory (positive weight) or inhibitory (negative weight). The input and output could be the digital or analog data values. Several processing units are interconnected according to a selected topology of the network to achieve a pattern recognition task.

The input of a processing unit may come from outputs of other processing units and or from an external source. The output of each unit may be given to several units including it. A network can be static or dynamic; some of the static networks use the back propagation algorithm and radial basis function with multilayer perceptrons. Some of the dynamical networks (recurrent networks) have output feedback, state feedback and feedforward dynamics.

The learning of the network can be supervised or unsupervised. In supervised learning, both inputs and outputs are presented to the network. In the unsupervised learning (self-recognizing networks), the inputs alone are presented to the network. Some of the algorithms for unsupervised learning are adaptive resonance theory (Carpenter and Grossberg, 1987) self organizing features maps (Kohonen, 1990). One of the important applications of ANN is in pattern recognition analysis. A pattern is a set of inputs and outputs. Either supervised or unsupervised training method can be used to train an ANN depending upon the network topology. In the supervised training, the difference between the calculated output of the network and the desired output of the pattern is minimized. To achieve the minimum difference, synaptic weights are updated. This procedure is adopted for all the patterns.

ALGORITHMS

Although, a number of algorithms are investigated, a sample number of algorithms are presented here and their performances are discussed.

Back Propagation Algorithm (BPA): BPA is one of the most studied and used algorithm for neural networks learning (Shihab, 2006). The BPA uses the Steepest Descent Method (SDM) to reach a global minimum. The SDM uses the error in the output layer of the network to update the weights of the network so as to reach the minimum of the objective function which is defined to the summation of squared error between the desired outputs and the network outputs. The algorithm uses a learning parameter called η. The algorithm works on supervised learning.

The number of iterations required for different values of η for different range of synaptic weights for SDM, the number of iterations required for constant weights for SDM and the number of iterations required for different hidden nodes with one hidden layer for SDM were found. A comparison is made between the iterations required for one hidden layer and two hidden layers in SDM, the iterations required for the nodes in the hidden layer for different value of η for SDM and the iterations required by the nodes in the hidden layer with and without θ in SDM were found.

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 (intrusion detection). Training indicates the formation of final weights which indicate a thorough learning of intrusion and normal packets along with corresponding labeling.

The convergence rate of BPA is shown in Fig. 1. The classification performance of BPA is shown in Table 1. In Table 2, false acceptance rate and false rejection rates are shown.

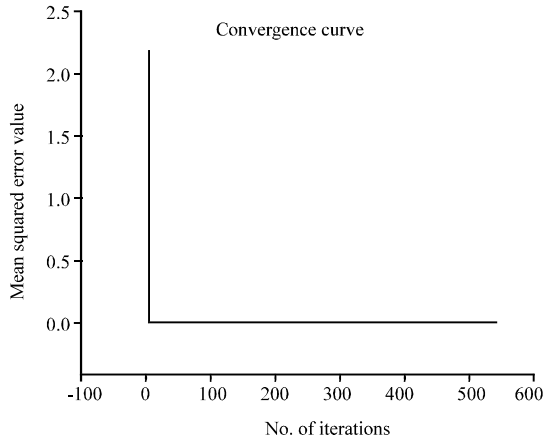

Fig. 1: Mean squared error curve

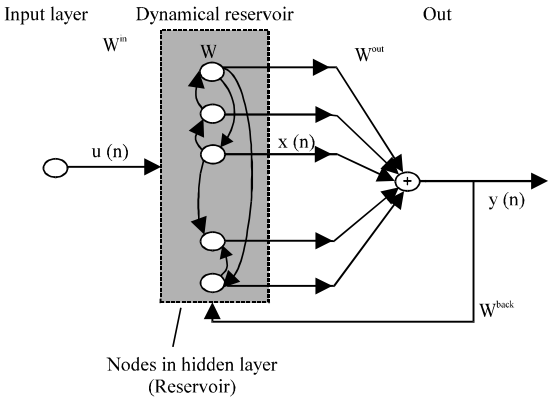

Fig. 2: An Echo State Neural Network (ESNN)

Table 1: Classification performance

Packet type	Total No. of tested	No. of classified	No. of misclassified
Normal	363	360	3
Intrusion	637	600	37

Table 2: False acceptance/rejection rate

Packet type	False Acceptance Rate (FAR)	False Rejection Rate (FRR)
Normal	5.8% (37/637)	0.8% (3/360)
Intrusion	10.1% (37/363)	0.4% (3/637)

Echo State Neural Network (ESNN): ESNN possesses a highly interconnected and recurrent topology of nonlinear PEs that constitutes a reservoir of rich dynamics and contains information about the history of input and output patterns (Jaeger, 2002a, b). The outputs of internal PEs (echo states) are fed to a memory less but adaptive readout network (generally linear) that produces the network output.

The interesting property of ESNN shown in Fig. 2 is that only the memory less readout is trained whereas the recurrent topology has fixed connection weights. This reduces the complexity of RNN training to simple linear regression while preserving a recurrent topology but obviously places important constraints in the overall architecture that have not yet been fully studied.

To train the ESNN, reservoirs and state matrix have to be used. The number of the iterations required for ESNN is lesser than the number of iterations required for SDM.

Training of ESNN: Training an ESN is a simple linear regression model task. In this, only the output activation function f^{out} is calculated while W^{in}, W and W^{back} never change after the initialization. The training is divided into the following three steps.

Network initialization:

- W^{in} and W^{back} are generated randomly
- Random sparse matrix W is generated and scaled to have a spectral radius of α where $\alpha<1$ to ensure the presence of echo states in the network

Sampling network training dynamics:

- Network inner units are initialized arbitrarily. For example, x (0) = 0
- The inner units states are updated for n = 0, 1, 2, ..., T using the equation:

$$x(n+1) = f(W^{in}u(n+1) + Wx(n) + W^{back}d(n)) \quad (1)$$

where d (t) is the teaching signal and d (0) = 0

- Network states before a washout time T_0 are ignored due to their dependency on the initial state
- Network states (u (n+1), x (n+1), d (n−1) after T_0 are collected in a state collecting matrix M of size $(T-T_0+1) \times (K+N+L)$
- $f^{out-1}(d(n))$ values after T_0 are collected in a teacher collecting matrix T of size $(T-T_0+1) \times L$

Computing output weights: Output weights are computed by evaluating the pseudoinverse matrix of M, multiplying it by T and then transposing it:

$$W^{out} = (M^{+} T)^{t}$$

Radial Basis Function (RBF): RBF have been found to be widely used for the interpolation of scattered data (Sarra, 2006). A Gaussian RBF monotonically decreases with distance from the centre. In contrast, a multiquadric RBF which in the case of scalar input monotonically increases with distance from the centre. Gaussian-like RBFs are local and are more commonly used than multiquadric-type RBFs which have a global response. Radial functions are simply a class of functions.

In principal, they could be employed in any sort of model (linear or nonlinear) and any sort of network (single-layer or multi-layer). RBF networks have traditionally been associated with radial functions in a single-layer network (Rapaka *et al*., 2003). The simulation of intrusion detection has been implemented. Table 3 shows the distribution of patterns chosen for training and testing. This data set has been separated using variance analysis into training (183 patterns) and testing (2973 patterns).

Table 3: Distribution of patterns chosen for training and testing

Class	Training pattern	Testing pattern
1 (normal)	148	1286
2 (snmpgetattack)	7	735
3 (smurf)	28	932
Total	183	2983

EXPERIMENTAL ANALYSIS

It is mandatory to use huge amount of patterns to be presented for training Echo State Neural Network (ESNN). However, it would take enormous amount of time for the ESNN to learn the patterns. Only 24 patterns have been considered for training purpose. Training indicates the formation of final weights which indicate a thorough learning of intrusion and normal packets along with corresponding labeling. Figure 3 shows the performance of the ESNN.

The simulation results were obtained from the standard KDD data set. It is a well defined as normal and with different types of attack for TCP, UDP and ICMP etc. A set of sample data set is shown in the study. Each row is a pattern. The fields in each pattern describe the properties of respective packet.

Sample KDD dataset

Packet details:

- 0, udp, private, SF, 105, 146, 0 , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 255, 254, 1.00, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, normal
- 0, udp, private, SF, 105, 146, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 255, 254, 1.00, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, normal
- 0, udp, private, SF, 105, 146, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 255, 254, 1.00, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, normal

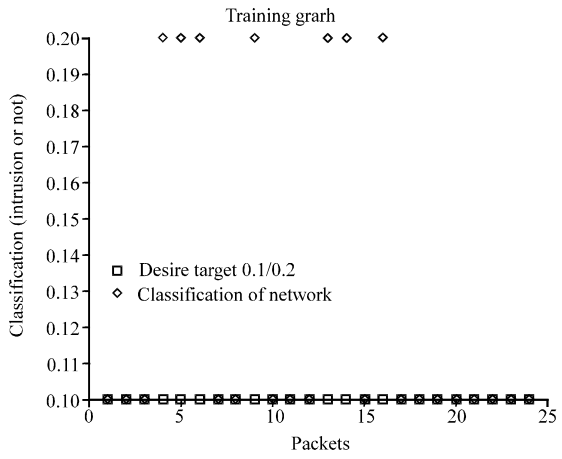

Fig. 3: Packet classification

Table 4: Sample dataset used for training

Patterns used for training input to ESNN after uncorrelating the features of patterns	Target outputs
0 .2 .01 .1 105 146 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0.00 0.00 0.00 1.00 0.00 0.00 255 254 1.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00	.1
0 .2 .01 .1 105 146 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0.00 0.00 0.00 1.00 0.00 0.00 255 254 1.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00	.1
0 .2 .01 .1 105 146 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0.00 0.00 0.00 1.00 0.00 0.00 255 254 1.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00	.1
0 .2 .01 .1 105 146 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0.00 0.00 0.00 1.00 0.00 0.00 255 254 1.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00	.2
0 .2 .01 .1 105 146 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0.00 0.00 0.00 1.00 0.00 0.00 255 254 1.00 0.01 0.01 0.00 0.00 0.00 0.00 0.00	.2

- 0, udp, private, SF, 105, 146, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 255, 254, 1.00, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, snmpget attack
- 0, udp, private, SF, 105, 146, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 255, 254, 1.00, 0.01, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00, snmpget attack

Instead of KDD data set, free sniffer softwares like network sniffer, packet sniffer and more softwares can be used to extract the values of a packet which can be further labeled as normal or an attack to be used for training. The contents of the packet should be suitably modified into meaningful numerical values. A sample dataset used for training all the three algorithms, back propagation algorithm, echostate neural network algorithm and radial basic function are shown in Table 4.

The topology of ESN used is 41×20×1; No. of nodes in the input layer is 41, No. of nodes in the hidden layer is 20 and No. of nodes in the output layer is 1. The labeling is set as 0.1 (normal) or 0.2 (attack). It is mandatory to use huge amount of patterns to be presented for training ESN. However, it would take enormous amount of time for the ESN to learn the patterns. Hence, only 24 patterns have been considered for training purpose.

The dataset has been separated as training and testing (intrusion detection). Training indicates the formation of final weights which indicate a thorough learning of intrusion and normal packets along with corresponding labeling. Table 5 shows number of patterns used for training and testing the performance of ESNN in classifying the intrusion packet. Table 6 shows number of patterns classified and misclassified.

Table 5: Distribution of patterns chosen for training

Packet type	Total number used for training
Normal	17
Intrusion	7

Table 6: Classification performance

Packet type	Total No. of tested	No. of classified	No. of misclassified
Normal	17	15	2
Intrusion	7	2	5

CONCLUSION

The study has been carried out to achieve faster and better performance from a set of already available information in the database. With the existing training and testing data, the classification performance is 100%. In this study, KDD dataset has been considered to experiment the performance of ESN in classifying the LAN intrusion packets. A topology of 41×20×1 had been chosen. The future research will involve in implementing an echo state neural network for classification of intrusion packet and suggested to implement other combinations of supervised and unsupervised ANN by incorporating additional intrusion data.

REFERENCES

Bu, S., F.R. Yu, X.P. Liu, P. Mason and H. Tang, 2011. Distributed combined authentication and intrusion detection with data fusion in high-security mobile ad hoc networks. IEEE Trans. Veh. Technol., 60: 1025-1036.

Carpenter, G.A. and S. Grossberg, 1987. ART 2: Self-organization of stable category recognition codes for analog input patterns. Applied Optics, 26: 4919-4930.

Chavan, S., K. Shah, N. Dave and S. Mukherjee, 2004. Adaptive neuro-fuzzy intrusion detection systems. Proceedings of the International Conference on Information Technology: Coding and Computing, April 5-7, 2004, Las Vegas, USA.

Helman, P. and G. Liepins, 1993. Statistical foundations of audit trail analysis for the detection of computer misuse. IEEE Trans. Software Eng., 19: 886-901.

Jaeger, H., 2002a. Short term memory in echo state networks. (Tech. Rep. No. 152). German National Research Center for Information Technology, Bremen.

Jaeger, H., 2002b. Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach. GMD Report 159, German National Research Center for Information Technology, pp: 48. http://minds.jacobs-university.de/sites/default/files/uploads/papers/ESNTutorialRev.pdf.

Katar C., 2006. Combining multiple techniques for intrusion detection. JCSNS Int. J. Comput. Sci. Network Secur., 6: 208-218.

Kohonen, T., 1990. The self-organizing maps. Proc. IEEE, 78: 1464-1480.

Kou, G., Y. Peng, Y. Shi and Z. Chen, 2006. Network intrusion detection by multi-group mathematical programming based classifier. Proceedings of the 6th IEEE International Conference on Data Mining Workshops, December 2006, Hong Kong, pp: 803-807.

Li, Q.H., S.Y. Jiang and X. Li, 2004. A supervised intrusion detection method. Proc. Int. Conf. Mach. Learn. Cybern., 3: 1475-1479.

Liu, J., F.R. Yu, C.H. Lung and H. Tang, 2009. Optimal combined intrusion detection and biometric-based continuous authentication in high security mobile ad hoc networks. IEEE Trans. Wireless Commun., 8: 806-815.

Rapaka, A., A. Novokhodko and D. Wunsch, 2003. Intrusion detection using radial basis function network on sequences of system calls. Int. Joint Conf. Neural Networks, 3: 1820-1825.

Sarra, S.A., 2006. Integrated multiquadric radial basis function approximation methods. Comput. Math. Appl., 51: 1283-1296.

Shihab, K., 2006. A back propagation neural network for computer network security. J. Comput. Sci., 2: 710-715.

Xue, J.S., J.Z. Sun and X. Zhang, 2006. Recurrent network in network in network intrusion detection system. Proceedings of the 3rd International Conference on Machine Learning and Cybernetics, August 26-29, 2006, Shanghai.

Zhang, B., X. Pan and J. Wang, 2007. Hybrid intrusion detection system for complicated network. Proceedings of the 4th International Conference on Fuzzy Systems and Knowledge, August 24-27, 2007, Haikou, Hainan, China.