International Journal of Advanced Research in
Computer Science Engineering and Information Technology
Volume: 4 Issue: 2 21-Feb-2015,ISSN_NO: 2321-3337

Implementation of Echo State Network for Intrusion Detection

S. Saravanakumar[1], R.Dharani[2]
Professor, Department of Information Technology, Panimalar Institute of technology[1]
Asst. Professor, Department of Information Technology, Panimalar Institute of technology[2]

ABSTRACT -*. 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 play a major role in detecting and disrupting various attacks before cooperating with the software. This work presents the investigations carried out on Echo State Network (ESN) structures for intrusion detection. New algorithms have been presented which have faster convergence and better performance in IDS from a set of available information in the database. This paper has been implemented with the KDD dataset to experiment the performance of ESN in classifying the Local Area Network (LAN) intrusion packets.*

Keywords−Echo State Network, Intrusion Detection, ANN, Malicious, DoS.

1, 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 software [16].

Every organization of even small size is connected to internet. 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 [15]. 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 [17].

Network-based intrusion detection, operate at the gateway of a network and examines all the incoming packets. Router-based intrusion detection is installed on 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 [11]. 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 [14], 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. The following are the classification of attack detection [13].

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. This 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, e.g., sudden appearance of never used protocols, big amount of unsuccessful login attempts.

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

Network Intrusion Detection System (NIDS) - This works on information provided by the network, mainly packets sniffed from the network layer [4][9]. This uses protocol decoding and heuristical analysis, and statistical anomaly analysis. This detects DoS with buffer overflow attacks, invalid packets, attacks on application layer and spoofing attacks.

The primary ways an intruder can get into the system is through primary intrusion, system intrusion and remote intrusion [2]. The IDS responses to set of actions when detects intrusions. Some of the responses are mentioned in [3], which involves reporting results and findings to a pre-specified location, while others are more active automated responses.

IDS can be viewed as the second layer of protection against unauthorized access to networked information systems because despite the best access control systems [1] and the intruders are still able to enter computer networks. IDS expand the security provided by the access control systems by using system administrators with a warning of the intrusion [8].

Different algorithms have been applied to model the various attack signatures and normal behavior response patterns of the systems. There are three commonly used algorithms used to model the various attacks, [10] and are named as, naive Bayes, Artificial Neural Network (ANN), and Decision Tree (DT).

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.

Normally, an 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 [17].

The importance of the present research work is to explore 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.

The rest of the paper is organized as follows. Section 2 describes the detailed implementation of the proposed echo state network architecture. Section 3 details the experimental setup and analysis, and finally conclusions are given in section 4.

2, Echo State Network

ESN is a novel approach to recurrent neural network training. An ESN consists of large fixed, recurrent reservoir network, from which the desired output is obtained by training suitable output connection weights. Determination of optimal output weights become a linear, uniquely solvable task of minimization. ESN provide a novel and easier way to manage approach to supervised training of Recurrent Neural Networks (RNNs) [6].

An ESN is a type of three-layered recurrent network with sparse, random, and crucially, untrained connections within the recurrent hidden layer. The ESN is a recurrent neural network with a sparsely connected hidden layer with typically 1% connectivity. The connection weights of the reservoir network are not changed by training. In order to compute a desired output dynamics, only the weights of connections from the reservoir to the output units are calculated [7].

ESN [7], [5] possesses a highly interconnected and recurrent topology of nonlinear Processing Elements (PEs) that constitutes a “reservoir of rich dynamics” and contains information about the history of input and output patterns. 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 ESN is shown in figure 1 and 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 ESN, reservoirs and state matrix have to be used. The number of the iterations required for ESN is lesser than the number of iterations required for SDM. Figure 2 shows an ESN with K input units and L output units.

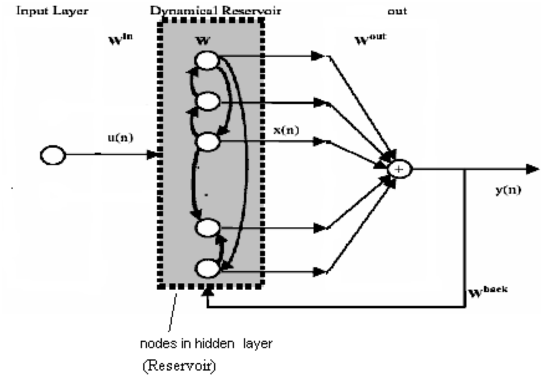

Figure 1 An echo state network

A. Tanhlike Activation Function

The activation function has to be applied to all the neurons in every iteration. This is important to choose an efficient function. Also at the same time choosing a suboptimal activation function can significantly degrade the ESNs output quality. The algorithm often used hyperbolic tangent, tanh(), that has high complexity requiring both large amounts of storage and a significant processing time. Because of these shortcomings, the following approximate function was developed [12].

$$TL(x) = \operatorname{sign}(x)\left[1 + \frac{1}{2^{\lfloor 2^n |x| \rfloor}}\left(\frac{2^n |x| - \lfloor 2^n |x| \rfloor}{2} - 1\right)\right] \quad (1)$$

B. Network Model

In the ESN model, each neuron, or unit of the network has an activation state at a given time step of 'n'. The network consists of a set of K input units with an activation vector u(n), a set of N inner units with an activation vector x(n), and a set of L output units with an activation vector y(n), as shown in figure 1. The network has a N x K input connection weight matrix W^{in}, a N x N internal connectivity matrix W, a L x (K + N) output weight matrix W^{out}, and optimal N x L output global feedback connection weight matrix W^{back}.

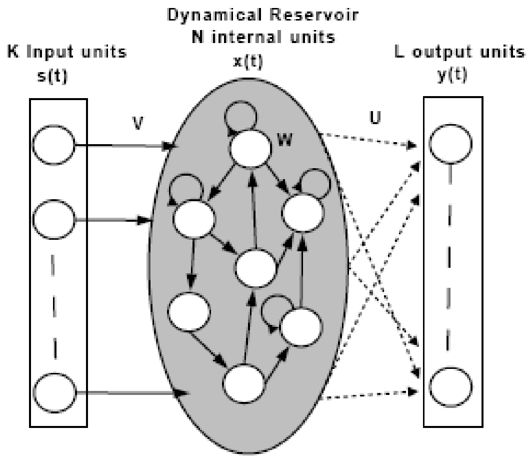

Figure 2 ESN with K input units

The activation states of the inner units are updated using the following equation.

$$x(n + 1) = f(W^{in}\, u(n + 1) + Wx(n) + W^{back}\, y(n)) \qquad (2)$$

where, *f* is the transfer activation function of the inner units. The output is calculated using the following equation.

$$y(n + 1) = f^{out}(W^{out}\, (u(n + 1), x(n + 1), y(n))) \qquad (3)$$

where, f^{out} is the output activation function and u(n + 1), x(n + 1), y (n) is the concatenation of the input, inner and output activation vectors. The hyperbolic tangent function (tanh) is usually used with *f* and f^{out}, though other sigmoidal and linear functions can be used as well.

C. Training of ESN

Training an ESN is a simple linear regression model task. In this, only 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.

1. 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

2. 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,

$$x(n + 1) = f(W^{in}\,u(n + 1) + Wx(n) + W^{back}\,d(n)) \quad (4)$$

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

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

3. Computing Output Weights

- Output weights are computed by evaluating the pseudoinverse matrix of M, multiplying it by T, and then transposing it.

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

3, Experimental Analysis

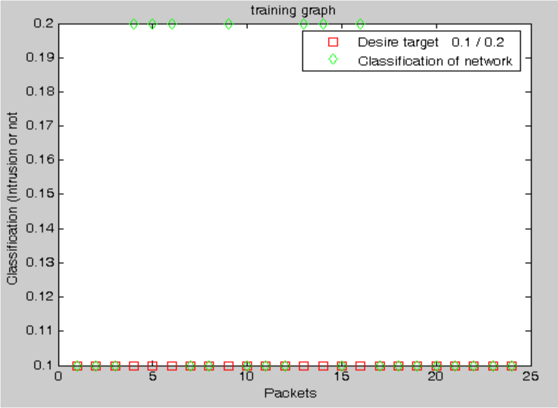

Figure 3 Packet classification

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, ICMP, etc. A set of sample data set is shown in Table 1. Each row is a pattern. The fields in each pattern describe the properties of respective packet.

Sl. No.	**Packet Details**
1	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.
2	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.
3	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.
4	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,snmpgetattack.
5	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,snmpgetattack.

Table 1 Sample KDD dataset

Sl. No	**Patterns used for training Input to ESNN after uncorrelating the features of Patterns**	**Target outputs**
1	0 .2 .01 .1 105 146 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	.1
2	0 .2 .01 .1 105 146 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	.1
3	0 .2 .01 .1 105 146 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	.1
4	0 .2 .01 .1 105 146 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	.2
5	0 .2 .01 .1 105 146 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	.2

Table 2 Sample dataset used for training

Instead of KDD data set, free sniffer software's like network sniffer, packet sniffer and more software's 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 is shown in Table 2.The topology of ESN used is 41 x 20 x 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 3 gives number of patterns used for training and testing the performance of ESNN in classifying the intrusion packet. Table 4 gives number of patterns classified and misclassified.

Packet Type	Total number used for training
Normal	17
Intrusion	7

Table 3 Distribution of patterns chosen for training

Packet Type	Total number tested	No. Classified	No. Misclassified
Normal	17	15	2
Intrusion	7	2	5

Table 4 Classification performance

4, Conclusions

The paper 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 paper, KDD dataset has been considered to experiment the performance of ESN in classifying the LAN intrusion packets. A topology of 41 x 20 x 1 had been chosen. The future work 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

1. Baojun Zhang, Xuezeng Pan, and Jiebing Wang, “Hybrid Intrusion Detection System for Complicated Network”, Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), 2007.
2. Gang Kou, Yi Peng, Yong Shi, and Zhengxin Chen, “Network Intrusion Detection by Multi-group Mathematical Programming based Classifier”, Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06), 2006.

3. Helman P., and Liepins G., “Statistical foundations of audit trail analysis for the detection of computer misuse”, IEEE Transaction on Software Engineering, Vol. 19, Issue 9, pp. 886-901, 1993.
4. Hu Zhengbing, Li Zhitang, and Wu Junqi, “A Novel Network Intrusion Detection System (NIDS) Based on Signatures Search of Data Mining”, Proceedings of the Workshop on Knowledge Discovery and Data Mining, 2008.
5. Jaeger H., “Short term memory in echo state networks”, Tech. Rep. No. 152, Bremen: German National Research Center for Information Technology, 2002.
6. Jaeger H., “The Echo State Approach to Analysing and Training Recurrent Neural Networks”, Technical Report- GMD Report 148, German National Research Center for Information Technology, 2001.
7. Jaeger H., “Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach”, GMD Report 159, Fraunhofer Institute AIS, 2002.
8. Jingg-Sheng Xue, Ji-Zhou Sun, and Xu Zhang, “Recurrent Network in Network in Network Intrusion Detection System”, Proceedings of the 3rd International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August, 2006.
9. John Zhong Lei, and Ali Ghorbani, “Network Intrusion Detection using an improved competitive learning neural network”, Proceedings of the second annual conference on Communication Networks and Services Research (CNSR ’04), 2004.
10. Katar C., “Combining multiple techniques for intrusion detection”, International Journal on Computer Science and Network Security (IJCSNS), Vol. 6, No. 2B, pp. 208-218, Feb. 2006.
11. Liu J., Yu F., Lung C. H., and Tang H., “Optimal combined intrusion detection and biometric-based continuous authentication in high security mobile ad hoc networks’, IEEE Transactions on Wireless Communication, Vol. 8, No. 2, pp. 806-815, February 2009.
12. Marra S., Iachino M., and Morabito F., “Tanh-like activation function implementation for high-performance digital neural systems”, Research in Microelectronics and Electronics 2006, pp. 237–240, June 2006.
13. Mishra A., Nadkarni K., and Patcha V. T. A., “Intrusion detection in wireless ad hoc networks”, IEEE Wireless Communication, Vol. 11, No. 1, pp. 48-60, Feb. 2004.
14. Moses Garuba, Chunmei Liu, and Duane Fraites, “Intrusion Techniques: Comparative Study of Network Intrusion Detection Systems”, Proceedings of the fifth International Conference on Information Technology: New Generations, 987-0-7695-3099-4/08 $25.00 ©, IEEE 2008.
15. Qing-Hua L, Sheng-Yi Jiang, Xin Li, “A Supervised Intrusion Detection Method”, Proceedings of the 3rd International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2004.
16. Sampada Chavan, Khusbu Shah, Neha Dave, and Sanghamitra Mukherjee, “Adaptive Neuro-Fuzzy Intrusion Detection Systems”, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC’04), 2004.
17. Shengrong Bu, Richard Yu F., Xiaoping P. Liu, Peter Mason, and Helen Tang, “Distributed Combined Authentication and Intrusion Detection with Data Fusion in

High-Security Mobile Ad hoc Networks”, IEEE Transactions on Vehicular Technology, Vol. 60, No. 3, pp. 1025-1036, March 2011.