Neural Network Approach for Intrusion Detection

Amit Kumar Choudhary
Department of Electrical Engineering
National Institute of Technology, Kurukshetra
Kurukshetra, Haryana, 136119, India
amitchoudhary005@yahoo.co.in

Akhilesh Swarup
Department of Electrical Engineering
Galgotias College of Engineering & Tech., Gr. Noida
On leave from NIT, Kurukshetra, India
aswarup@nitkkr.ac.in

ABSTRACT

Intrusion Detection System is based on the belief that an intruder's behavior will be noticeably different from that of a legitimate user and would exploit security vulnerabilities. This paper proposes a neural network approach to improve the alert throughput of a network and making it attack prohibitive using IDS. For evolving and testing intrusion the KDD CUP 99 dataset are used. The result of proposed approach is found to be more efficient in the area of Intrusion Detection and promises a good scope for further research.

Keywords

Generalized Regression Neural Network. Intrusion Detection System, Neural Network.

1. INTRODUCTION

Intrusion is defined as an unauthorized access to a malicious activity on a computer or an information system that violates the security policy. Intrusion detection (ID) is the process used to identify intrusions that is identifying individuals who are using a computer system without authorization (i.e., crackers) and those who have legitimate access to the system but are exceeding their privileges (i.e., the insider threat). It is based on the beliefs that an intruder's behavior will be noticeably different from that of a legitimate user. Intrusion detection has been an active field of research for about two decades, starting in 1980 with the publication of John Anderson's Computer Security Threat Monitoring and Surveillance [1] followed by Dorothy Denning's seminal paper, "An Intrusion Detection Model," [2] published in 1987, providing a methodological framework that inspired many researchers and laid the groundwork for commercial products. The analysis relies on sets of predefined rules that are provided by an administrator or created by the system. With the proliferation of networks, it is natural to extend single-host IDSs to network IDSs. Unfortunately, extending intrusion detection to small local networks and then to a wide area network (WAN), is not sufficient in today's heavily interconnected environment. The problem lies in the fact that the intruder is an intelligent and flexible agent while the rule-based IDSs obey fixed rules.

Different detection techniques are employed to search for attack patterns in the design of IDSs, namely misuse detection model looking for the exploitation of known weak points in the system, which can be described by a specific pattern or sequence of events or data. Next, is an anomaly detection model detecting change in the patterns of utilization or behavior of the system. Fig.1 gives a computer network with Intrusion Detection System installed on it. Still, despite of substantial research and commercial investments, ID technology is immature and its effectiveness is limited. This problem can be tackled by the application of soft computing techniques in IDSs.

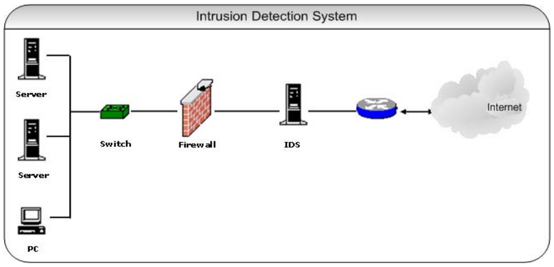

Figure 1. A Computer Network with Intrusion Detection System

Soft computing is a general term for describing a set of optimization and processing techniques that are tolerant of imprecision and uncertainty to achieve robustness and low solution cost. Based on the principal constituents of soft computing techniques as Fuzzy Logic (FL), Artificial Neural Networks (ANN), Probabilistic Reasoning (PR), and Genetic Algorithm (GA) number of approaches has been proposed for detecting network intrusions [3, 4]. The idea behind the application of soft computing techniques is to include an intelligent agent in the system that is capable of disclosing the latent patterns in abnormal and normal connection audit records, and to generalize the patterns to new (and slightly different) connection records of the same class. Successful IDSs can recognize both intrusions and denial-of-service activities and invoke countermeasures against them in real time.

ICIS 2009, November 24-26, 2009 Seoul, Korea
Copyright © 2009 ACM 978-1-60558-710-3/09/11... $10.00"

Section I of this paper introduced the basic ideas in intrusion detection. Section II motivates for the IDS. Section III deals with intrusion detection systems and its basics. Section IV deals with the dataset, attack types, and the features used for classifying network connection records. Section V gives the introduction of neural network in the area of intrusion detection and presents an overview of some of the previous studies that have applied neural networks in intrusion detection. Section VI describes the methodology and training-validation method and also presents the experimental results. Section VII concludes the paper with a discussion of the results and possibilities for future work. Last two sections presents the acknowledgement and the references.

2. MOTIVATION

There are number of enemies to the network security namely hackers, unaware disgruntled staff and snoopers. They create computer programs like viruses, Trojan horse programs, vandals and spam thus, harming the network security in one or the other ways. Thus, there has been an extensive choice of technologies, ranging from anti-virus software packages to dedicated security hardware such as firewalls. Organizations continue to deploy firewalls as their central gatekeepers and are increasingly looking to additional security technologies to counter risk. An intrusion detection system provides around-the-clock network surveillance analyzing and searching for unauthorized activity. In the physical analogy, an IDS is equivalent to a video camera and motion sensor; detecting unauthorized or suspicious activity and working with automated response systems, such as watch guards, to stop the activity. Thus, Intrusion detection is a viable and practical approach for providing a different notion of security in today's huge and existing infrastructure of computer and network systems.

3. INTRUSION DETECTION SYSTEM

An Intrusion Detection System is a computer program that attempts to perform ID by either misuse or anomaly detection, or a combination of techniques. An IDS should preferably perform its task in real time.

3.1 Desirable Characteristics of an IDS

As the number of systems to be monitored increases and the chances of attacks increase we also consider the following characteristics as desirable. Firstly, it must run continually with minimal human supervision and must be fault tolerant that is, it must be able to recover from system crashes and reinitializations. Next, it must resist subversion as must be able to monitor itself and detect if it has been modified by an attacker. Then it must impose a minimal overhead on the system where it is running. It must be able to be configured according to the security policies of the system that is being monitored. They must be able to scale to monitor a large number of hosts. It must allow dynamic reconfiguration, this is, the ability to reconfigure the IDS without having to restart it.

The main task of intrusion detection systems is defense of a computer and network system by detecting an attack and possibly repelling it. Once an intrusion has been detected, IDS issues alerts notifying administrators of this fact. The next step is undertaken either by the administrators or the IDS itself.

3.2 Location of Intrusion Detection Systems in Network

Usually an intrusion detection system captures data from the network and applies its rules to that data or detects anomalies in it. Depending upon the network topology, the type of intrusion activity (i.e. internal, external or both), and our security policy (what we want to protect from hackers), IDSs can be positioned at one or more places in the network. Fig. 2 shows typical locations where an intrusion detection system can be placed.

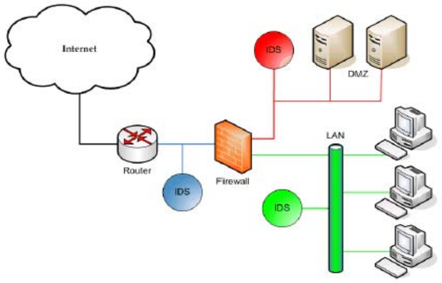

Figure 2. Typical locations where an IDS should be placed

4. DATASET, ATTACK TYPES & THEIR FEATURE

To evaluate the performance of proposed real-time IDS system, we use Knowledge Discovery in Database (KDD) Cup 99 [5] (the raw training data obtained by simulating a typical U.S. Air Force LAN for seven weeks) supplied by the Defense Advanced Research Projects Agency (DARPA) and the Massachusetts Institute of Technology's Lincoln Labs in 1998 [6]. For each TCP/IP connection, 41 various quantitative and qualitative features were extracted. Some describe the connection itself and rest describes the properties of connections to the same host in last two seconds. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. The four different categories of attack patterns are:

A. Denial of Service (DOS) Attacks: It's a class of attacks in which an attacker makes some computing or memory resource too busy to handle legitimate requests, or denies legitimate users access to a machine. Examples are Back, Ping of death and Smurf.

B. User to Superuser or Root Attacks (U2Su or U2R): In these classes of attacks, an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system. Examples are Eject, Perl and Xterm.

C. Remote to User Attacks (R2L): It's a class of attacks in which an attacker sends packets to a machine over a network to gain local access as a user of that machine. Examples are Dictionary, Guest and Imap.

D. Probing: It is a class of attacks in which an attacker scans a network of computers to gather information or find known vulnerabilities. Examples are Mscan, Nmap and Satan.

This paper, tries to improve the detection process with the implementation of soft computing technique particularly ANN with respect to the previously obtained result that used a dataset wit two attack types namely Neptune and Satan to a great extent.

5. NEURAL NETWORK INTO INTRUSION DETECTION

The ability of soft computing techniques for dealing with uncertain and partially true data makes them attractive to be applied in intrusion detection. However, Artificial Neural Networks (ANNs) are the most commonly used soft computing technique in IDSs [4,9,10].

An ANN is an information processing system that is inspired by the way biological nervous systems. The most important property of a Neural Network is to automatically learn / retrain coefficients according to data inputs and data outputs. Applying the Neural Network approach to Intrusion Detection, we first have to expose NN to normal data and attacks to automatically adjust coefficients of the NN during the training phase. Performance tests are then conducted with real network traffic and attacks.

Neural Networks have been largely employed with success for complex problems such as Pattern Recognition, Hand-written Character Recognition, and Statistical Analysis. Some recent studies on the application of the Neural Network approach to the scope of Intrusion Detection are stated here. Ryan, Lin, and Miikkulainen [7] described an off-line anomaly detection system, which utilized a back-propagation MLP neural network. Then Cannady [8] of Georgia Technical Research Institute (GTRI) conducted a research to apply Multi-Level Perceptron (MLP) model and Self-Organizing Maps (SOM) for misuse detection. The final result succeeded in classification of normal and attack records in 89-91% of the cases. In yet another study, Cunningham and Lippmann [9] of the MIT Lincoln Laboratory used a MLP model for misuse detection. With the Neural Network approach, false alarms were reduced and the detection rate increased to roughly 80% with the DARPA database. Another study by Mukkamala [10], described the three and four layer neural networks and reported results of about 99.25% correct classification for their two class (normal and attack) problem. This paper is aimed to solve an off-line multi class problem using regression method in which not only the attack records are distinguished from normal ones, but also the attack type is identified. The promising results of the present study show the potential applicability of ANNs for developing high efficiency practical IDSs.

6. METHODOLOGY AND EXPERIMENTAL ANALYSIS

The Generalized Regression Neural Network (GRNN) paradigm has been proposed [11] as an alternative to the popular back-propagation training algorithm for feedforward neural networks. It is closely related to the probabilistic neural network [12]. Regression can be thought of as the least-mean-squares estimation of the value of a variable based on available data. The GRNN is based on the estimation of a probability density function from observed samples using Parzen window estimation [13]. It utilizes a probabilistic model between the independent vector random variable X with dimension D, and dependent scalar random variable Y. Assume that x and y are the measured values for X and Y variables, respectively. If $f(x,Y)$ represents the known joint continuous probability density function, and if $f(x,Y)$ is known, then expected value of Y given x (the regression of Y on x) can be estimated as

$$E[Y \mid x] = \frac{\int_{-\infty}^{\infty} Yf(x,Y)dY}{\int_{-\infty}^{\infty} f(x,Y)dY} \tag{1}$$

Based on p sample observations that are available, i.e., on the training set given by x and Y, further assuming that the underlying density is continuous and the first partial derivatives of the function evaluated at any x are small, the probability estimator $\hat{f}(x,y)$ can be written as

$$\hat{f}(x,y) = \frac{1}{(2\pi)^{D+1/2}\sigma^{D+1}}\frac{1}{p}\times \sum_{i=1}^{p}\left[\exp\left(-\frac{(x-x_i)^T(x-x_i)}{2\sigma^2}\right)\exp\left(-\frac{(y-y_i)^2}{2\sigma^2}\right)\right] \tag{2}$$

Where x_i and y_i are the ith training set data, and x_i denotes the vector form of variable x. A physical interpretation of the probability estimate $\hat{f}(x,y)$ is that it assigns a sample probability of width σ for sample x_i and y_i, after that, the probability estimate is the sum of those sample probabilities.

Substituting (2) into (1), the desired conditional mean of Y given x, $\hat{y}$, can be calculated as

$$\hat{y}(x) = E[Y \mid x] = \sum_{i=1}^{n}[y_i \exp(d_i)] / \sum_{i=1}^{n}\exp(d_i) \tag{3}$$

where d_i is given by the distance function of the input space.

Now let us consider each element of the vector K, namely k_i, to be estimated by an individual GRNN. If the weighted average approach is used to construct the output of GRNN, then each k_i can be written as

$$\hat{k}_i = \frac{\sum_{j=1}^{m}[k_j \exp(d_j)]}{\sum_{j=1}^{m}\exp(d_j)} \tag{4}$$

Where d_j, the distance function and here can be written as

$$d_j = \left[-\left(\frac{s - s_j}{\sigma} \right)^2 \right] \qquad (5)$$

In the above expression s is the new input and s_j is the stored input, σ is the spread factor. In (4) k_j is the stored output corresponding to s_j and $\hat{k}_i$ implies the estimated value of true k_i. Now $\hat{k}_i$ is the output of the GRNN and a good estimation of k_i depends on the selection of spread factor σ. This paper defines s_j as [1 0 0] for normal conditions, [0 1 0] for Neptune attack and [0 0 1] for the Satan attack.

Different possible values for selected features were extracted from the protocol used and a numerical value was attributed to each of them. For example, the possible numerical values attributed are: tcp=0, udp=1, icmp=2. These numerical representations are necessary because the feature vector fed to the input of the neural network has to be numerical. Some of the features have binary values and some others have a continuous numerical range (such as duration of connection). As a result, the features are normalized by mapping all the different values for each feature to [0, 1] range. The accuracy of classification problems depends on a variety of parameters, ranging from the architecture of the actual neural network to the training algorithm of choice.

Moradi and Zulkernine [4], were the first to implement intrusion detection with a three layer MLP (two hidden layers with 35 neurons in each) referred to as: {35 35 35 3}. At this stage, early stopping validation was not applied and the training was performed for 200 times and the process took more than 25 hours. Then he used 900 datasets (300 of each type) and trained it. It took almost 5hours because of the implementation of early stopping validation method. With this, the training error decreased at the 45th epochs that can be seen in fig 3. Figure shows the mean square error of the back propagation training process versus the progress of training epochs. The correct classification rate was more than 90% showing an 11% increase from the former experiment.

A GRNN is used for function approximation in the present study. It has a radial basis layer and a special linear layer. The symbolic representation has been used to express each of the three conditions in such a way that, a "1" in a column indicates the occurrence of the column's corresponding string and a "0" indicates a non-occurrence. The best two layer neural network used in this study was {41 41 41}. The best result was attained in a training session that was stopped on 5th epoch. The result was 100% correct classification on training and 100% on the testing set shown in the fig 4, giving a more accurate performance compared to the result reported earlier [4]. This is a preliminary result with a static data.

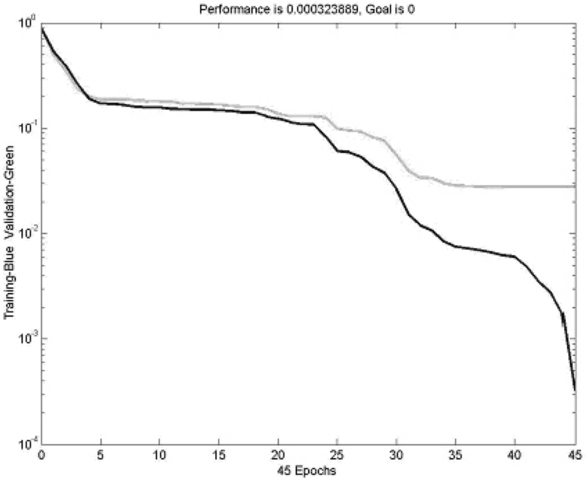

Figure 3. The training process error when the early stopping validation method is applied. The darker curve shows the error on the training set and the brighter curve presents the error on validation set.

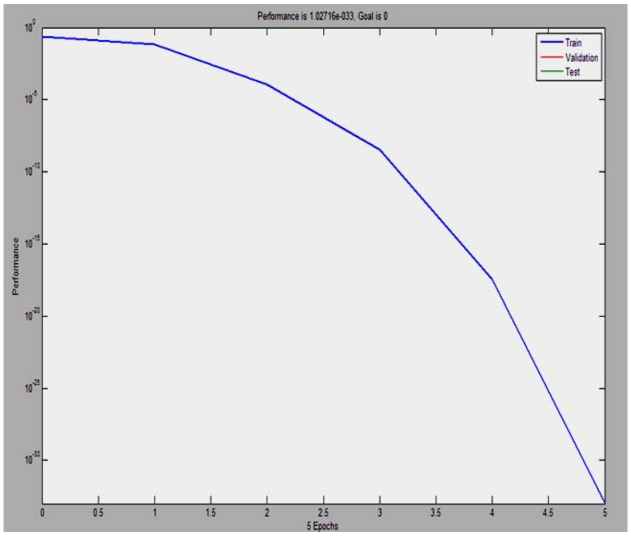

Figure 4. The training process error when GRNN is applied. Here, validation and training dataset coincide each other (darker part).

In a previous study [10], a result of more than 99% correct classification on this dataset using the neural network structure {41-40-40-1} was reported. In another similar study with different dataset [8], the success rate was comparable to the results of the present study (89-99%) and again a two class problem was implemented.

7. CONCLUSION

This paper presents an intrusion detection system based on neural networks. The implemented neural network model was used to solve a three-class problem, that is, normal, attack patterns, and the type of the attack. When given data is presented to the model, the results obtained revealed a great deal of accuracy app. 100%. As a possible future development to the present study, one can include more attack scenarios in the dataset. Practical IDSs should include several attack types. In order to avoid unreasonable complexity in the neural network, an initial classification of the connection records to normal and general categories of attacks can be the first step. The records in each category of intrusions can then be further classified to the attack types.

8. ACKNOWLEDGMENT

The authors would like to thank to Mr. Moradi and Mr. Zulkernine from Queen's University Kingston, Ontario, Canada, for presenting their work in this domain and thus, the authors used their ideas and dataset to obtain a new and improved result.

9. REFERENCES

[1] J.P. Anderson, "*Computer Security Threat Monitoring and Surveillance*", tech. report, James P. Anderson Co., Fort Washington, Pa., 1980.

[2] D. E. Denning, "*An intrusion detection model*," IEEE Transactions on Software Engineering, vol. 13, no. 2, pp. 222–232, 1987.

[3] S. M. Bridges and R. B. Vaughn, "*Fuzzy Data Mining And Genetic Algorithms Applied To Intrusion Detection*", Proceedings of 12th Annual Canadian Information Technology Security Symposium, pp. 109-122, 2000.

[4] M. Moradi, and M. Zulkernine, "*A Neural Network Based System for Intrusion Detection and Classification of Attacks*," IEEE International Conference on Advances in Intelligent Systems - Theory and Applications, Luxembourg-Kirchberg, Luxembourg, November 15-18, 2004.

[5] University of California at Irvine, 1999. KDD Cup: <http://kdd.ics.uci.edu/databases/kddcup99/task.htm>.

[6] MIT Lincoln Laboratory, 1999 DARPA intrusion detection evaluation: design and procedures, DARPA Technical Report 1062, http://www.ll.mit.edu. Feb. 2001.

[7] J. Ryan, M. Lin, and R. Miikkulainen, "*Intrusion Detection with Neural Networks*," AI Approaches to Fraud Detection and Risk Management: Papers from the 1997 AAAI Workshop, Providence, RI, pp. 72-79, 1997.

[8] James Cannady, "*Artificial neural networks for misuse detection*," Proceedings of the 1998 National Information Systems Security Conference (NISSC'98), Arlington, VA, 1998.

[9] R. Cunningham and R. Lippmann, "*Improving intrusion detection performance using keyword selection and neural networks*," Proceedings of the International Symposium on Recent Advances in Intrusion Detection, Purdue, IN, 1999.

[10] Srinivas Mukkamala, "*Intrusion detection using neural networks and support vector machine*," Proceedings of the 2002 IEEE International Honolulu, HI, 2002.

[11] D. F. Specht, "*A general regression neural network*," IEEE Trans. Neural Network, vol. 2, no. 6, pp. 568–576, 1991.

[12] M. Timothy, "*Advanced Algorithms for Neural Networks: AC++ Sourcebook*," Wiley, Canada, 1995.

[13] E. Parzen, "*On estimation of a probability density function and mode*," Ann. Math. Statist.33 pp. 1065–1076, 1962