International Journal of Computer Applications Technology and Research
Volume 3– Issue 9, 585 - 588, 2014, ISSN: 2319–8656

Intrusion Detection System Using Self Organizing Map Algorithms

Faezeh Mozneb khodaie
Department of computer,
Shabestar branch, Islamic Azad
University, Shabestar, Iran

Mohammad Ali Jabraeil Jamali
Department of computer,
Shabestar branch, Islamic Azad
University, Shabestar, Iran

Ali Farzan
Department of computer,
Shabestar branch, Islamic Azad
University, Shabestar, Iran

Abstract: With the rapid expansion of computer usage and computer network the security of the computer system has became very important. Every day new kind of attacks are being faced by industries. Many methods have been proposed for the development of intrusion detection system using artificial intelligence technique. In this paper we will have a look at an algorithm based on neural networks that are suitable for Intrusion Detection Systems (IDS). The name of this algorithm is "Self Organizing Maps" (SOM). So far, many different methods have been used to build a detector that Wide variety of different ways in the covers. Among the methods used to detect attacks in intrusion detection is done, In this paper we investigate the Self-Organizing Map method.

Keywords: Intrusion Detection System; Self Organizing Maps; Attacks; Security; neural network

1. INTRODUCTION

The goal of intrusion detection is to discover unauthorized use of computer systems. Existing intrusion detection approaches can be divided into two classes - anomaly detection and misuse detection. Anomaly detection approaches the problem by attempting to find deviations from the established patterns of usage. Misuse detection, on the other hand, compares the usage patterns to known techniques of compromising computer security. Architecturally, an intrusion detection system can be categorized into three types – hostbased IDS, network-based IDS and hybrid IDS. Host-based IDS, deployed in individual hostmachines, can monitor audit data of a single host. Network-based IDS monitors the traffic data sent and received by hosts. Hybrid IDS uses both methods.

Self-Organizing Map has been successfully applied in complex application areas where traditional method has failed. Due to their inherently non-linear nature, they can handle much more complex situations than the traditional methods. One of those problems represents intrusion detection by intrusion detection systems. These systems deal with high dimension data on the input, which is needed to map to 2-dimension space. Designed architecture of the intrusion detection system is application of neural network SOM in IDS systems. Over the last few decades information is the most precious part of any organization. Most of the things what an organization does revolve around this important asset. Organizations are taking measures to safeguard this information from intruders. The rapid development and expansion of World Wide Web and local networks and their usage in any industry has changed the computing world by leaps and bounds [1][2].

2. INTRUSION DETECTION SYSTEMS

Intrusion Detection System is a system that identifies , in real time, attacks on a network and takes corrective action to prevent them. They are the set of techniques that are used to detect suspicious activity both at network and host level. There are two main approaches to design an IDS.

1) Misuse Based Ids (Signature Based)
2) Anomaly Based Ids.

In a misuse based intrusion detection system , intrusions are detected by looking for activities that correspond to know signatures of intrusions or vulnerabilities [3]. While an anomaly based intrusion detection system detect intrusions by searching for abnormal network traffic . The abnormal traffic pattern can be defmed either as the violation of accepted thresholds for frequency of events in a connection or as a user's violation of the legitimate profile developed for normal behavior.

One of the most commonly used approaches in expert system based intrusion detection systems is rule-based analysis using Denning's profile model [3]. Rule-based analysis depends on sets of predefined rules that are provided by an administrator. Expert systems require frequent updates to remain current. This design approach usually results in an inflexible detection system that is unable to detect an attack if the sequence of events is slightly different from the predefined profile [4]. Considered that the intruder is an intelligent and flexible agent while the rule based IDSs obey fixed rules . This problem can be tackled by the application of soft computing techniques in IDSs. Soft computing is a general term for describing a set of optimization and processing techniques. The principal constituents of soft computing techniques are Fuzzy Logic (FL), Artificial Neural Networks (ANNs), Probabilistic Reasoning (PR), and Genetic Algorithms (GAs) [4].

3. TYPES OF NETWORKING ATTACKS

There are four major categories of networking attacks. Every attack on a network can be placed into one of these groupings [4].

3.1 Denial of Service (DoS):

A DoS attacks is a type of attack in which the hacker makes a memory resources too busy to serve legitimate networking requests and hence denying users access to a machine e.g. apache, smurf, Neptune, ping of death, back, mail bomb, UDP storm, etc.

3.2 Remote to User attacks (R2L):

A remote to user attack is an attack in which a user sends packets to a machine over the internet, and the user does not have access to in order to expose the machines vulnerabilities and exploit privileges which a local user would have on the computer, e.g. xlock, guest, xnsnoop, phf, sendmail dictionary etc.

3.3 User to Root Attacks (U2R): These attacks are exploitations in which the hacker starts off on the system with a normal user account and attempts to abuse vulnerabilities in the system in order to gain super user privileges, e.g. perl, xterm.

3.4 Probing: Probing is an attack in which the hacker scans a machine or a networking device in order to determine weaknesses or vulnerabilities that may later be exploited so as to compromise the system. This technique is commonly used in data mining, e.g. satan, saint, portsweep, mscan, nmap etc.

4. SELF ORGANIZING MAP

The Self-Organizing Map [5] is a neural network model for analyzing and visualizing high dimensional data. It belongs to the category of competitive learning network. The SOM Figure 1. defines a mapping from high dimensional input data space onto a regular two dimensional array of neurons.

In designed architecture is input vector with six input values and output is realized to 2 dimension space. Every neuron i of the map is associated with an n dimensional reference vector $m_i\left[m_1,........,m_n\right]^T$, where n denotes the dimension of the input vectors. The reference vectors together form a codebook. The neurons of the map are connected to adjacent neurons by a neighborhood relation, which dictates the topology, or the structure, of the map. Adjacent neurons belong to the neighborhood Ni of the neuron i. In the SOM algorithm, the topology and the number of neurons remain fixed from the beginning. The number of neurons determines the granularity of the mapping, which has an effect on the accuracy and generalization of the SOM. During the training phase, the SOM forms an elastic net that is formed by input data. The algorithm controls the net so that it strives to approximate the density of the data. The reference vectors in the codebook drift to the areas where the density of the input data is high. Eventually, only few codebook vectors lie in areas where the input data is sparse.

The learning process of the SOM goes as follows:

1. One sample vector x is randomly drawn from the input data set and its similarity (distance) to the codebook vectors is computed by using Euclidean distance measure:

$$\| x - m_c \| = \min_i\{\| x - m_i \|\}$$

2. After the BMU has been found, the codebook vectors are updated. The BMU itself as well as its topological neighbors are moved closer to the input vector in the input space i.e. the input vector attracts them. The magnitude of the attraction is governed by the learning rate. As the learning proceeds and new input vectors are given to the map, the learning rate gradually decreases to zero according to the specified learning rate function type. Along with the learning rate, the neighborhood radius decreases as well. The update rule for the reference vector of unit i is the following:

$$m_i(t+1) = m_i + a(t)h_{ci}(r(t))[x(t) - m_i(t)]$$

3. The steps 1 and 2 together constitute a single training step and they are repeated until the training ends. The number of training steps must be fixed prior to training the SOM because the rate of convergence in the neighborhood function and the learning rate are calculated accordingly.

After the training is over, the map should be topologically ordered. This means that n topologically close input data vectors map to n adjacent map neurons or even to the same single neuron.

4.1 Mapping precision

The mapping precision measure describes how accurately the neurons respond to the given data set. If the reference vector of the BMU calculated for a given testing vector xi is exactly the same xi, the error in precision is then 0. Normally, the number of data vectors exceeds the number of neurons and the precision error is thus always different from 0. A common measure that calculates the precision of the mapping is the average quantization error over the entire data set:

$$E_q = \frac{1}{N}\sum_{i=1}^{N}\left\|x_{i+}m_c\right\|$$

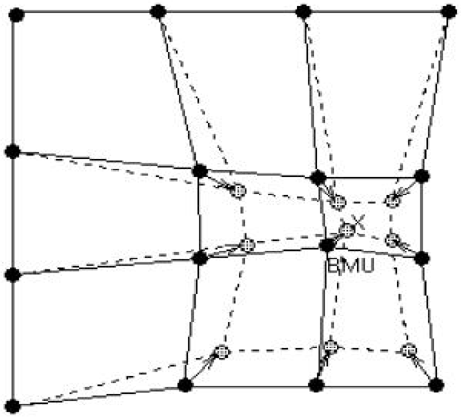

Figure. 1 General SOM topology

2.2 Topology preservation

The topology preservation measure describes how well the SOM preserves the topology of the studied data set. Unlike the mapping precision measure, it considers the structure of the map. For a strangely twisted map, the topographic error is big even if the mapping precision error is small.

A simple method for calculating the topographic error:

$$E_q = \frac{1}{N}\sum_{i=1}^{N}u_x(x)$$

where $u(x_k)$ is 1 if the first and second BMUs of x_k are not next to each other. Otherwise $u(x_k)$ is 0.

5. THE ARCHITECTURE SELF-ORGANIZING MAP METHOD

The Self-Organizing Map method is mapped to the data Normal initially trained Then the mixture of normal and attack data to be tagged. After this step, the experimental data to give mapped To determine whether the input vector The normal vector or a vector of attack. If that BMU selected is a normal neuron with labels In this case the normal vector of the detected Otherwise traffic in general, the attack is detected [6]. Figure 2. shows the architecture.

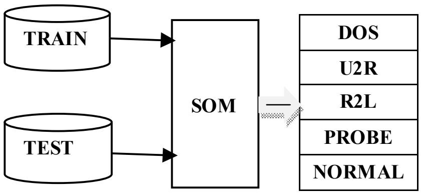

Figure. 2 Architecture Self-Organizing Map method

Training data set in this data set is a mixture of normal and attack. Table 1. the number and type of data in the training data set shows. The size of the test data sets are shown in Table 2.

Table 1. The number of data vectors in the training data set

Attack Type	Count
Dos	45927
U2R	52
R2L	995
Probe	11656
Normal	26944

Table 2. The number of data vectors in the data set to test the type of attack

Attack Type	Count
Dos	5741
U2R	37
R2L	2199
Probe	1106
Normal	9711

5. RESULTS AND EVALUATION CRITERIA

To simulate the Self-Organizing Map of the simulation tool box in MATLAB is used for Self-Organizing Map (SOM TOOLBOX, 2012). Evaluation criteria used are as follows:

5.1 Total Error

Percentage of The total number of errors made The data have been trained And test data.

5.2 False Positive

Event normal system as an attack is detected. This event is not an attack, but the attack was seen. When we tested the data And compare them with data from the trained If you have been attacked, and the attack has been detected This is an error.

5.3 True Negative

Activities or events without risk of That have been labeled as normal activity. The event of an attack, but the attack has not been seen.

Table 3. shows the results of the evaluation on the basis of these criteria indicates the number of error found.

Table 3. Evaluation results show that the Self-Organizing Map based on these criteria.

Method	Self-Organizing Map Method	
Criteria	Error Count	Accuracy
Total Error	9384	50.07
False Positive	9384	0.06
True Negative	0	0

6. CONCLUSIONS

The Self Organizing Map is an extremely powerful mechanism for automatic mathematical characterization of acceptable system activity. In the above paper we have described how we can use Self Organizing Maps for building an Intrusion Detection System. We have explained the system architecture and the flow diagram for the SOMe We have also presented the pros and cons of the algorithm.

The results show that Algorithms used in the Self-Organizing Map method gives the optimal solutions to large amounts of data. The Self-Organizing Map method is trained only with normal data Thus, errors can not be calculated for each type of attack.

7. REFERENCES

[1] Damiano Bolzoni, Sandro Etalle, Pieter H. Hartel, andEmmanuele Zambon. Poseidon: a 2-tier anomaly-based networkintrusion detection system. In Proceedings of the 4th IEEE International Workshop on Information

Assurance, 13-14 April 2006, Egham, Surrey, UK, pages 144-156, 2006.

[2] D. A. Frincke, D. Tobin, 1. C. McConnell, 1. Marconi, and D. Polla. A framework for cooperative intrusion detection. In Proc. 21st NIST-NCSC National Information Systems SecurityConference, pages 361-373, 1998.

[3] Denning D, "An Intrusion-Detection Model", IEEE Transactionson Software Engineering, Vol. SE-13, No 2, Feb 1987.

[4] Simon Haykin, "Neural Networks: A ComprehensiveFoundation", Prentice Hall, 2nd edition, 1999.

[5] Kohonen, T. 1995. Self-Organizing Maps, volume 30 of Springer Series in InformationSciences. Berlin, Heidelberg: Springer. (SecondExtended Edition 1997).

[6] Kohonen T., Oja E., Simula O., Visa A., Kangas J., , "Engineering applications of the self-selforganizing map.", Proceedings of the IEEE, Vol. 84, Issue: 10, Pages: 1358 – 1384, 1996.