ANOMALY INTRUSION DETECTION DESIGN USING HYBRID OF UNSUPERVISED AND SUPERVISED NEURAL NETWORK

M. Bahrololum, E. Salahi and M. Khaleghi

IT Security and Systems Group
Iran Telecommunication Research Center, Tehran, Iran.

bahrololum, e_salahi, khaleghi@itrc.ac.ir

ABSTRACT

This paper proposed a new approach to design the system using a hybrid of misuse and anomaly detection for training of normal and attack packets respectively. The utilized method for attack training is the combination of unsupervised and supervised Neural Network (NN) for Intrusion Detection System. By the unsupervised NN based on Self Organizing Map (SOM), attacks will be classified into smaller categories considering their similar features, and then unsupervised NN based on Backpropagation will be used for clustering. By misuse approach known packets would be identified fast and unknown attacks will be able to detect by this method.

KEYWORDS

Intrusion Detection System, Self Organizing Map, Backpropagation, Neural Network, Anomaly detection

1. INTRODUCTION

As the threat becomes a serious matter year by year, intrusion detection technologies are indispensable for network and computer security. Intrusion Detection is the process of monitoring events in a system or network to determine whether an intrusion occurred or not. An Intrusion Detection System (IDS) monitors network traffic for suspicious activity and alerts the system or network administrator against malicious attacks [1]. The primary aim of IDS is to protect the availability, confidentiality and integrity of critical networked information systems.

Two main approaches to intrusion detection system are used namely misuse and anomaly detection [2]. Misuse detection is based on a description of known malicious activities. This description is often modelled as a set of rules referred to as attack signatures. An anomaly detection IDS looks for anomalies, meaning it thinks outside of the ordinary. It uses rules or predefined concepts about “normal” and “abnormal” system activity (called heuristics) to distinguish anomalies from normal system behavior and to monitor report on, or block anomalies as they occur.

Various artificial intelligence techniques have been utilized in anomaly IDS, such as machine learning [3-4], data mining, pattern recognition and neural networks [5].

At the first we designed an unsupervised Neural Network based Intrusion Detection for Normal packets, and then we employed supervised Neural Network based on Backpropagation in the system for clustering and classifying of network attacks.

The used data in our experiments originates from MIT's Lincoln Lab; a well known dataset. It was developed for intrusion detection system evaluations by DARPA and is considered a benchmark for IDS evaluations [6]. We perform experiments to classify the network traffic patterns according to 5-class taxonomy. The five classes of patterns in the DARPA data are normal, probe, denial of service, user to super-user, and remote to local.

This paper is organized as follows: Section 2 briefly introduces two models of neural network, SOM (unsupervised NN) and Backpropagation (supervised NN). Section 3 proposes our designing method. Section 4 presents the details about KDD cup 99 dataset used for training and detection the Intrusion Detection system. Section 5 summarizes the obtained results.

2. Neural Network Models

2.1. Kohonen Self Organizing Map

Kohonen's self organizing maps (SOMs) have become a firmly technique in cluster analysis. SOMs are adapted by unsupervised learning. The unsupervised learning process in SOM can be briefly described in three steps. In first step the connection weights are assigned with small random numbers at the beginning and choosing the learning rate parameter. On step 2, best matching unit is determined, the method of determining the winner uses the squared Euclidean distance between the input vector and the weight vector and chosen the unit whose weight vector has the smallest Euclidean distance from the input vector. In the last stage, the weights are updated following Kohonen's learning rule according to the formula 1. The weight update only occurs for the active output neurons. Typically, the unit whose weight vector was closet to the input vector is allowed to learn.

$$\omega_{ij}(new) = \omega_{ij}(old) + \alpha\left[x_i - \omega_{ij}(old)\right] \qquad (1)$$

Where xi is the ith input vector, ωij is the jth column of the weight matrix, and α, the learning rate, decreases as learning proceeds.

This training process continues until all input vectors are processed. Convergence criterion utilized here is epochs, which defines how many times all input vectors should be fed to the SOM for training.

2.2. Backpropagation

Backpropagation is a supervised learning method of teaching artificial neural networks. The aim of backpropagation is to train the net to achieve a balance between the ability to respond correctly to the input patterns that are used for training (memorization) and the ability to give reasonable responses to input that is similar, to that used in training according figure 1. The training of a network by backpropagation involves three stages: The feed forward of the input training pattern, the calculation and backpropagation of the associated error, and the adjustment of the weights, so that the forward pass produces an output vector for a given input vector based on the current state of the network weights. Since the network weights are initialized to random values, it is unlikely that reasonable outputs will result before training. The weights are adjusted

to reduce the error by propagating the output error backward through the network. This process is where the backpropagation neural network gets its name and is known as the backward pass:

* Compute error values for each node in the output layer.

* Compute the error for the middle layer nodes.

* Adjust the weight values to improve network performance using the Delta rule.

* Compute the overall error to test network performance.

The training set is repeatedly presented to the network and the weight values are adjusted until the overall error is below a predetermined tolerance. Since the Delta rule follows the path of greatest decent along the error surface, local minima can impede training. The momentum term compensates for this problem to some degree.

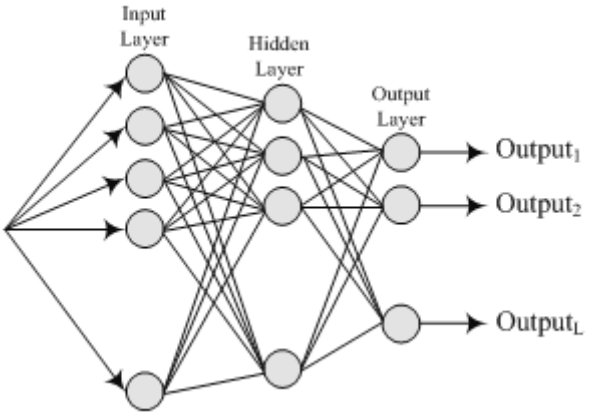

Figure 1. Backpropagation Neural Network

3. DESIGNING THE PROPOSED METHOD

This system is process of identifying the abnormal and normal packets in the network that are two phases. The first is the training phase that trained by hybrid the SOM and Backpropagation NN. Next phase is detection phase. This method is shown in Figure 2.

Since the operations of normal packets are specified and they show expected behavior, we could use the knowledge based (misuse) IDS detection, while unexpected activity (presumably an intrusion would be unusual) is continually designed and progressed and could not be seen as a knowledge based attack, therefore the anomaly IDS detection is performed over attacks. By the unsupervised NN based on Self Organizing Map (SOM), attacks will be classified into smaller categories considering their similar features, and then unsupervised NN based on Backpropagation will be used for clustering

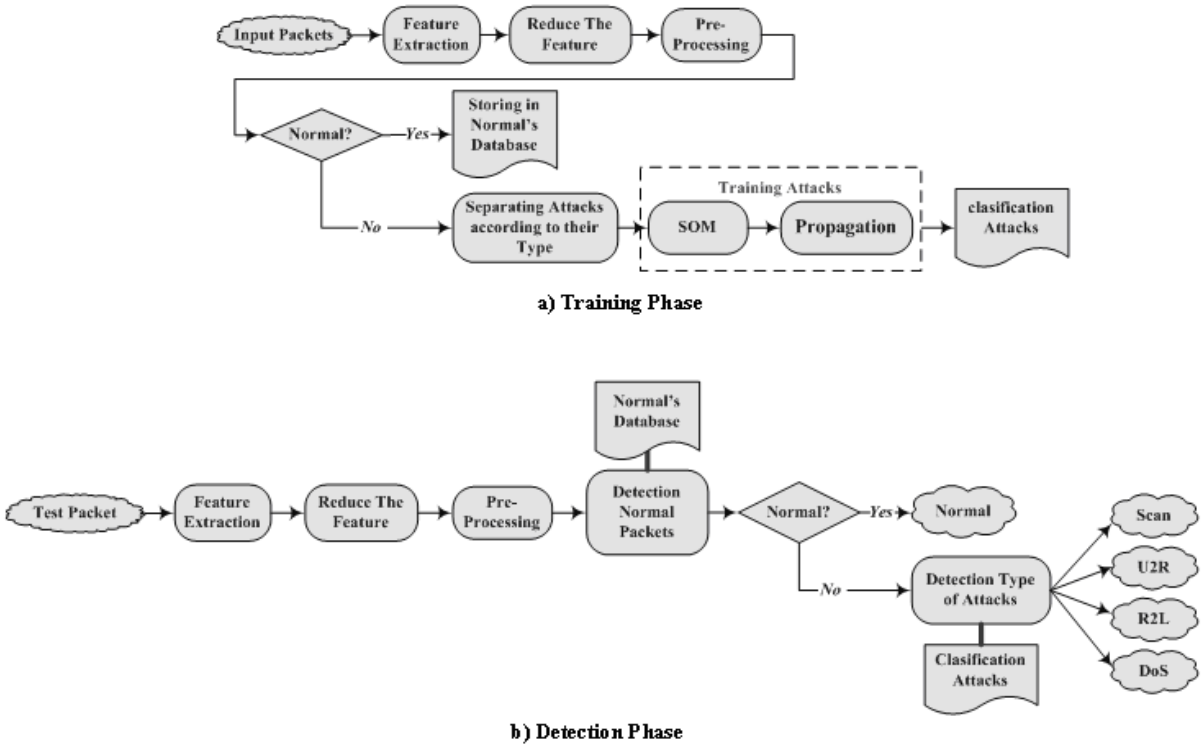

Figure 2. Designing the proposed method

3-1. Training Phase

- **Feature Extraction**

A connection in the KDD-99 dataset is represented by 41 features, each of which is in one of the continuous, discrete and symbolic form, with significantly varying ranges. The features in columns 2, 3, and 4 in the KDD99 dataset are the protocol type, the service type, and the flag, respectively. The value of the protocol type may be tcp, udp, or icmp; the service type could be one of the 66 different network services such as http and smtp; and the flag has 11 possible values such as SF or S2. Other information in these connection are length of the connection; number of data bytes from source to destination and vice versa; number of connections to the same host as the current connection in the past two seconds, etc. A complete listing of the set of features defined for the connection records is given in [7].

- **Reducing Features and Pre-Processing**

Derived features will be reduced from each of network packets, since may be irrelevant with poor prediction ability to the target patterns, and some of the them may be redundant due to they are highly inter-correlated with one of more of the other features which decreases not only the detection speed but also detection accuracy possibly [8].

After reducing KDD features from each record, pre-processing will be done by converting each feature from text or symbolic into numerical form. In this conversion, for each symbol an integer code is assigned. For instance, in the case of protocol type feature, 0 is assigned to tcp, 1

to udp, and 2 to the icmp symbol. Attack names were first mapped to one of the five classes, 0 for Normal, 1 for Probe, 2 for DoS, 3 for U2R, and 4 for R2L.

Three features spanned over a very large integer range, namely length [0, 60000], src_bytes [0, 1.3 billion] and dst_bytes [0, 1.3 billion]. Logarithmic scaling (with base 10) was applied to these features to reduce the range to [0.0, 4.78], [0.0, 9.14] and [0.0, 9.14] respectively. All other features were boolean, in the range [0.0, 1.0].

For normalizing feature values, a statistical analysis is performed on the values of each feature based on the existing data from KDD Cup's 99 dataset and then acceptable maximum value for each feature is determined.

3-2. Testing Phase

Similar to the training phase, features would be extracted and the same bits removed according to the first phase. The pre-processing will be performed. After it, the normal packets would recognized by the same database which used for keeping the normal data. Otherwise input packet recognized as attack and will be clustered in the most compatible group according to classification attacks in the training phase.

4. Intrusion Detection Training

The first requirement of each evaluating system is a set of input data for processing and determining the security level. We trained and tested our system using KDD Cup's 99 dataset.

4-1. Evaluation Dataset

The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate research in IDSs. A standard set of data to be audited, was provided and called "DARPA dataset". This includes a wide variety of intrusions simulated in a military network environment. The 1999 KDD intrusion detection contest uses a version of this dataset [6]. Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a LAN simulating a typical U.S. Air Force LAN.

DARPA dataset is separated into two categories, testing dataset and Training dataset. The raw testing dataset was TCP dump data from two weeks of network traffic. This was processed into about two million connection records. These connection records are not labeled. The raw training dataset was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic. This was processed into about five million connection records. Each connection record is labeled as either normal, or as an attack, with exactly one specific attack type that mention in below. In the training dataset there are 23 different attack types, according to Table 1 and in the testing dataset there are 37 different attack types according to Table 2.

4-2. Types of Attacks in Dataset

The raw dataset was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records. We choose almost 310,000 records from this database as the test set. Each connection record consists of about 100 bytes.

In KDD99, Attacks fall into four main categories [9]:

Table 1. Attacks in KDD99's Training dataset

Classification of Attacks	Attack Name
Probing	Port-sweep, IP-sweep, Nmap, Satan
Denial of Service (DoS)	Neptune, Smurf, Pod, Teardrop, Land, Back, Apache2
User to Root (U2R)	Buffer-overflow, Load-module, Perl, Rootkit, spy
Remote to Local (R2L)	Guess-password, Ftp-write, Imap, Phf, Multihop, Warezmaster, Warezclient

Table 2. Attacks in KDD99's Testing dataset

Classification of Attacks	Attack Name
Probing	Port-sweep, IP-sweep, Nmap, Satan, **Saint, Mscan**
Denial of Service (DoS)	Neptune, Smurf, Pod, Teardrop, Land, Back, Apache2, **Udpstorm, Process-table, Mail-bomb**
User to Root (U2R)	Buffer-overflow, Load-module, Perl, Rootkit, **Xterm, Ps, Http-tunnel, Sql-attack, Worm, Snmp-guess**
Remote to Local (R2L)	Guess-password, Ftp-write, Imap, Phf, Multihop, Warezmaster, **Snmpgetattack, Named, Xlock, Xsnoop, Send-mail**

- **Probing:** is a class of attacks where an attacker scans a network to gather information or find known vulnerabilities; surveillance and other probing, e.g., port scanning.
- **DOS:** Denial of Service is a class of attacks where an attacker makes some computing or memory resource too busy or too full to handle legitimate requests, thus denying legitimate users access to a machine, e.g. Syn-flood.
- **U2R:** User to root exploits are a class of attacks where 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; unauthorized access to local super user (root) privileges, e.g., various “buffer overflow” attacks.
- **R2L:** A remote to user attack is a class of attacks where an attacker sends packets to a machine over a network, then exploits machine’s vulnerability to illegally gain local access as a user; unauthorized access from a remote machine, e.g. guessing password.

The number of attacks in train and Test set for each of the elements is according to table 3.

Table 3. KDD Cup 99 Training and Testing Set

Attack Type	DDOS	Probe	U2R	R2L	Total Attack	Normal
Training Set	391458	4107	52	1126	494020	97277
Testing Set	229853	4166	2636	13781	311029	60593

4-3. Experimental Result

- Table 4 shows the Confusion Matrix obtained for SOM IDS for all features.
- The top left entry in the confusion matrix shows that 56945 of the actual "normal" test examples were predicted to be normal by this entry. The last column indicates that in total 93.98% of the "normal" examples were recognized correctly.
- The bottom row shows that 73.47% of test examples said to be normal were indeed "normal" in reality. The false positive rate for Normal class is 100-73.47 =26.53%.
- Simulation results are presented in Table 5. From the results it can be seen that the Performance of proposed method IDS for four categories of data is significant compared to SOM classifier.

Table 4. Confusion Matrix for the SOM Technique

Predicated Actual	Normal	Probe	DoS	U2R	R2L	Correct
Normal	56945	3448	190	8	2	93.98
Probe	1200	2679	280	5	2	64.30
DoS	7964	996	220889	3	1	96.10
U2R	101	68	9	49	1	21.49
R2L	11295	2993	7	0	1894	11.7
correct	73.47	21.30	99.78	75.38	99.68	

Table 5. Confusion Matrix for Proposed Method

Predicated Actual	Normal	Probe	DoS	U2R	R2L	Correct
Normal	59303	454	727	0	109	97.87
Probe	445	2985	639	0	97	71.65
DoS	6022	827	222958	0	46	97.00
U2R	198	16	9	0	5	0
R2L	11850	9	9	0	4321	26.69
correct	76.21	69.56	99.38	0	94.38	

5. Conclusion

Due to applying misuse method for normal packets identification, the processing time will decrease. Moreover using both unsupervised and supervised neural network for attacks identification causes better training and able to recognize unknown attacks. Experimental results in standard dataset KDD99 prove that this method is able to achieve accuracy better than SOM.

References

[1] H. Lin, C. Jiang, H. Huang, "A Secure and Efficient Model for Network Defensive Systems", ICS Dec 2006, Taiwan.

[2] P. Garcı´a-Teodoro, J. Dı´az-Verdejo, G. Macia´-Erna´ndez, E. Va´zquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges", computers & security, Vol. 28, pp. 18-28, 2009.

[3] S. Mukkamala, G. Janoski, A. Sung, "Intrusion detection using neural networks and support vector machines" Proceedings of the 2002 IEEE International Joint Conference on Neural Networks, pp. 1702 – 1707.

[4] M. Sabhnani and G. Serpen, "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context", The 2003 International Conference on Machine Learning; Models, Technologies and Applications, pp. 209-215.

[5] Z. Zhang and C. Manikopoulos, "Neural Networks in Statistical Anomaly Intrusion Detection", Journal of Neural Network World, Vol. 3, 2001, pp. 305-316.

[6] Lincoln Laboratory, Massachusetts Institute of Technology (MIT), 1998-2000. DARPA Intrusion Detection Evaluation, KDD dataset; http://kdd.ics.uci.edu//databases/kddcup98/kddcup98.html.

[7] Srilatha Chebrolu et.al, "Feature deduction and ensemble design of intrusion detection systems", Elsevier Journal of Computers & Security" Vol. 24/4, pp. 295-307, 2005.

[8] Y. Chen, A. Abraham, and J. Yang, "Feature Selection and Intrusion Detection Using Hybrid Flexible Neural Tree", LNCS 3498, pp. 439-444, 2005.

[9] Richard Lippmann, Joshua W. Haines, David J. Fried, Jonathan Korba, Kumar Das, "The 1999 DARPA off-line intrusion detection evaluation", The International Journal of Computer and Telecommunications Networking, Vol. 34, pp. 579-595, 2000.