Enhancement of Intrusion Detection Using Back Propagation Algorithm

Khattab M. Ali Alheeti
Information System Department
College of Computer - University of Anbar

المستخلص :

يعتبر كشف الاختراق من القضايا المهمة لحماية نظم المعلومات. وهذا البحث يهتم باختيار الخاصية المهمة التي تدخل في بناء نظام كشف الاختراق. ولتبسيط المشكلة يمكن استبعاد الخواص الغير مهمة او عديمة الفائدة وهذا يؤدي الى تسريع عملية الكشف وبنفس الوقت الحصول على نتائج افضل وبدقة عالية وبالتالي يكون عمل ترتيب الخواص من الامور المهمة في كشف الاختراق وباستخدام الشبكات العصبية سيكون الاداء أفضل وأكثر دقة حيث نقوم بتطبيق تقنية حذف واحد من الخواص في كل وقت وإجراء التجارب على الشبكة العصبية ونرى مدى تاثيرة على دقة الكشف الاختراق وعلى هذا الأساس نقوم بترتيب الخواص حسب الأهمية وهذه الخواص التي تدخل هي مجموعة من قواعد البيانات (DARPA)، والتي سوف تدخل تصنف الدخيل إلى خمس فئات من فئات الاختراق. وباستخدام الشبكات العصبية كأساس في بناء نظام كشف وبخواص اقل سنحصل على أداء عالي وكفائة جيدة.

Abstract :

Intrusion detection is an important component of secure information systems. This paper concerns the issue of identify main input features in building an intrusion detection system (IDS). Since exclusion of the unimportant and/or useless inputs leads to a simplification of the problem, faster and more accurate detection may result. Feature ranking and selection, therefore, is an important issue in intrusion detection. Since Neural network (NNs) tend to scale better and run

faster than other with higher accuracy, we apply the technique of deleting one feature at a time to perform experiments on NNs to rank the importance of input features for the DARPA collected intrusion data. Important features for each of the five classes of intrusion patterns in the data set are identified. It is shown that NN-based IDSs using a reduced number of features can deliver enhanced or comparable performance.

Keywords: Network security, intrusion Detection, Neural networks.

I. Introduction

This paper mainly addresses the issue of identifying important input features for intrusion detection. Science the ability to identify the important inputs and redundant inputs of a classifier leads directly to reduced size, faster training and possibly more accurate results, it is critical to be able to identify the important features of network traffic data for intrusion detection in order for the IDS to achieve maximal performance. The data we used in our experiments originated from MIT's Lincoln Lab. It was developed for intrusion detection system evaluations by DARPA and is considered a benchmark for intrusion detection evaluations [7]. We performed experiments to rank the importance of input features for each of the five classes (Normal, Probe, DOS, U2R, and R2L) of patterns in the DARPA data. It is shown that using only the important features for classification gives good accuracies and, in certain cases, reduces the training time and testing time of the (NNs) classifier.

II. Introduction to Intrusion Detection System

Intrusion detection systems (IDS) were proposed to complement prevention-based security measures. An intrusion is defined to be a violation of the security policy of the system; intrusion detection thus refers to the mechanisms that are developed to detect violations of system security policy [4]. Intrusion detection is an important component of a network's security system. It complements existing security technologies (such as firewalls) by providing crucial information to the network administrators about attacks and intrusions that may be undetected by existing security technologies. They also provide important information that will allow organizations to trace back the origins of attacks and aid in the prosecution of the attackers. Intrusion detection systems (IDS) have traditionally been classified into two categories *anomaly* detection and *misuse*- or signature-based detection. Misuse detection systems match incoming network traffic to a database of known attack signatures to detect intrusions. While a misuse detection system enjoys a high rate of success at detecting known attacks, they are ineffective at detecting new or unknown attacks. On the other hand, anomaly detection systems create a normal profile of the network or host under observation and flag deviations from the normal profile as probable intrusions. As these systems predict anomalous behavior, they have the advantage of being able to detect new and novel attacks [1].

However, IDS's have not kept pace with the rapidly evolving field of computer networking. An example is the domain of high speed networks especially gigabit networks, where the large amount of network data that the network produces is posing new challenges in anomaly detection.

Prohibitively large volume of network data makes the tasks of storing, classifying, and labeling the data almost infeasible. We can, of course, obtain labeled data by simulating intrusions in a network. However, then we would be limited to the set of known attacks and we would not be able to detect new attacks. As a result, it has been seen that currently available commercial solutions to detect intrusions in gigabit networks can detect less than half of the attacks directed at them [2] at gigabit speeds. The motivation behind our intrusion detection structure is simple: sampling reduces the number of features that needs to be processed, Thereby enabling anomaly detection in high-speed networks. In typical cases, sampling would lead to loss of information, leading to inaccurate predictions and/or faise alarms. To avoid such a state, the proposed predicative model to detect the intruder with low number of the feature at the same time high rate accuracy. Hence we will use in this system proposed *best eight features* select from data set, 41 attributes that describe the different features of the corresponding connection (22 of these features describe the connection itself and 19 of them describe the properties of connections to the same host in last two seconds).

III. Artificial Neural Networks (ANNS) in Intrusion Detection

The ability of soft computing techniques for dealing with unsure and partly true data makes them attractive to be applied in intrusion detection. Some studies have used soft computing techniques other than ANNs in intrusion detection. For example, genetic algorithms have been used along with decision trees to automatically generate rules for classifying network connections [3]. However, ANNs are the most commonly used soft computing technique in IDSs [11], [8], [6], [16], and [14]. An ANN is an information processing system that is inspired by the way biological nervous systems, such as the brain, process information. It is composed of a large number of highly interconnected processing elements (neurons) working with each other to solve specific problems. Each processing element (neuron) is basically a summing element followed by an activation function. The output of each neuron (after applying the weight parameter associated with the connection) is fed as the input to all of the neurons in the next layer. The learning process is essentially an optimization process in which the parameters of the best set of connection coefficients (weighs) for solving a problem are found and includes the following basic steps [15]:

- Present the neural network with a number of inputs (vectors each representing a pattern) - Check how closely the actual output generated for

a specific input matches the desired output.

- Change the neural network parameters (weights) to better approximate the outputs. Some IDS designers exploit ANN as pattern recognition technique. Pattern recognition can be implemented by using a feed-forward neural network that has been trained accordingly. During training, the neural network parameters are optimized to associate outputs (each output represents a class of computer network connections, like normal and attack (DOS, Prob, U2R, R2L)) with corresponding input patterns (every input pattern is represented by a feature vector extracted from the characteristics of the network connection record). When the neural network is used, it identifies the input pattern and tries to output the corresponding class. When a connection record that has no output associated with it is given as an input, the neural network gives the output that corresponds to a taught input pattern that is least different from the given pattern [6]. The most commonly reported application of neural networks in IDSs is to train the neural net on a sequence of information units, each of which may be an audit record or a sequence of commands. The input to the net consists of the current command and the past *w* commands (*w* is the size of window of commands under examination). Once the net is trained on a set of representative command sequences of a user, it constitutes (learns) the profile of the user and when put in action, it can discover the variance of the user from its profile [15], [10].

A. Backpropagation technique for intrusion detection

Backpropagation is a neural network learning algorithm. A neural network is a set of connected units following a particular topology. Each neuron is described by a unit that has an input and an output. Two neurons are connected if the output of one of them is connected to the input of the other. Each connection in a neural network has a weight associated to it. The topology of the neural network, the training methodology for weights' adjustment and the connections between the different neurons define the type of the corresponding neural network. In our paper, we are interested in the multilayer neural networks using the Backpropagation learning algorithm [5]. In a multilayer neural network, there are three kinds of layers. Each layer contains a set of neurons. The first layer, called *input layer*, sets the activation of its neurons according to the provided pattern in question. The *output layer* provides the answer of the network. A multilayer network may contain one or many *hidden layers* although in practice, usually one is used. Like any supervised learning technique, a multilayer neural network has two phases. The learning phase where the network learns by adjusting the

weights so as to be able to predict the correct class label of the new input patterns during the test phase. Before the training process, one should define the number of hidden layers (if more than one) and the number of neurons on each layer. The number of neurons on the input layer corresponds to the number of attributes that represent a sample. However, input values should be numerical to perform the backpropagation algorithm. Therefore, the discrete values are transformed into a vector as it is explained in the following. For each different discrete value of an attribute is assigned a neuron on the input layer. One output unit, on the output layer, may be used to represent exactly one class. So, if the output of a neuron on the output layer is equal to 1 then the corresponding class is designed as the predicted class. The number of hidden layers and the number of units on each hidden layer is established by experience during the training phase since there are no clear rules as to set the *best* number of hidden layer units. The use of neural networks in intrusion detection is not new because there are at least two works that were developed during the last decades. The first model is used in hyper view [6] for a user behavior modeling. The second one is that discussed in [12]. This latter was used as a misuse detection tool where only packet header attributes are considered for analysis to detecting denial of service and port scan attacks. While these works used neural networks for either user anomaly detection or misuse detection, we use them here for both network misuse and anomaly detection particularly over the different KDD 2000 data sets [9].

B. Importance of data reduction for intrusion detection systems

IDSs have become essential and generally used tools for ensuring network security. Since the quantity of audit data that an IDS needs to examine is very large even for a small network, classification by hand is impossible. Analysis is difficult even with computer assistance because extraneous features can make it harder to detect suspicious behavior patterns. Complex relationships exist between the features, which are practically impossible for humans to discover. An IDS must therefore reduce the amount of data to be processed. This is extremely important if real-time detection is desired. Reduction can occur in one of several ways Data that are not considered useful can be filtered, leaving only the potentially interesting data. Data can be grouped or clustered to reveal hidden patterns. By storing the characteristics of the clusters instead of the individual data, overhead can be significantly reduced. Finally, some data sources can be eliminated using feature selection.

IV. The Data Set

In the DARPA KDD cup 2000 intrusion detection evaluation program, an environment was set up to

acquire raw TCP/IP dump data for a network by simulating a typical U.S. Air Force LAN. The LAN was operated like a true environment, but being blasted with multiple attacks. For each TCP/IP connection, 41 various quantitative and qualitative features were extracted. Of this database a subset of 494021 data were used, of which 20% represent normal patterns [13].

Attack types fall into four main categories:

1. DOS: denial of service.

2. R2L: unauthorized access from a remote machine.

3. U2R: unauthorized access to local super user (root) privileges.

4. Probing: surveillance and other probing.

V. Ranking the Significance of Inputs

Feature selection and ranking is an important issue in intrusion detection. Of the large number of features that can be monitored for intrusion detection purpose, which are truly useful, which are less significant, and which may be useless? The question is relevant because the elimination of useless features (or audit trail reduction) enhances the accuracy of detection while speeding up the computation, thus improving the overall performance of an ID. In cases where there are no useless features, by concentrating on the most important ones we may well improve the time performance of IDS without affecting the accuracy of detection in statistically significant ways.

A. Data filtering

The purpose of data filtering is to decrease the amount of data processed by the IDs. Some data may not be useful to the IDS and thus can be eliminated before processing. This has the benefit of decreasing storage space requirements, reducing processing time and improving the detection rate (as data irrelevant to intrusion detection are discarded). However, filtering may throw out useful data, and so must be done carefully.

B. Feature selection

In difficult classification domains, some data may hinder the classification process. Features may contain false correlations, which hinder the process of detecting intrusions. Further, some features may be redundant since the information they add is contained in other features. Extra features can increase computation time, and can impact the accuracy of IDS [3]. Feature selection improves classification by error and trial for the subset of features, which best classifies the training data. The features under consideration depend on the type of IDS, for example, network-based IDS will analyze network related information such as packet destination IP address, logged in time of a user, type of protocol, duration of

connection etc. It is not known which of these features are redundant or irrelevant for IDS and which ones are relevant or essential for IDS. There does not exist any model or function that captures the relationship between different features or between the different attacks and features. If such a model did exist, the intrusion detection process would be simple and straightforward. In this paper we use the ANNs for feather selection. The subset of selected features is then used to detect intrusions.

VI. The Proposed Model IDS using Neural Networks

We selected the important features using the Artificial Neural Networks. After training and testing of each property we note the total number of features that ***eight*** of which have more influence in the accuracy of intrusion detection, form the ANN's of the class node as explained in Section '*Importance of data reduction for intrusion detection systems*'. These *8*-variables are *C, E, F, L, W, X, Y, and AB,* and can be observed in the appendix. Furthermore, Back propagation network classifier was constructed using the training data and then the classifier was used on the test data set to classify the data as an attack (Five classes) or normal data.

A. Algorithm:

Model starts with the following

i. Collect Data Set 2000 from DRAPA.
ii. Data Set encoding.
iii. Uniform Selection.
iv. Normalization.
v. Training and test data, ANNs.
vi. Then determine what features is the most effect.

The following figure (1) illustrates the work of the proposed model and a series of actions that will occur on the data set before the training and testing:

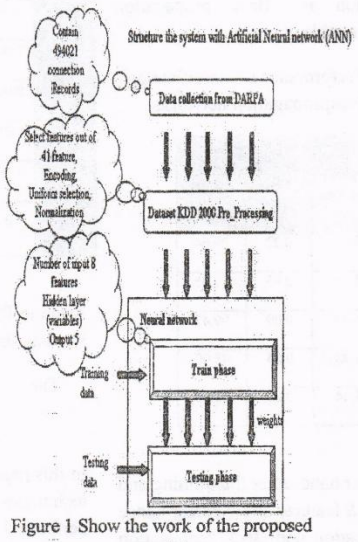

Figure 1 Show the work of the proposed model

The following figure (2) shows the categories of the model:

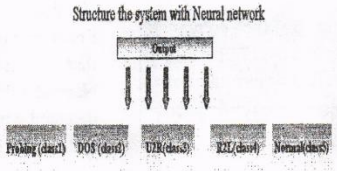

Figure 2 Show the class of the output.

VII. Results

After the training and testing of all the *41* features demonstrate Performance of classification and Back propagation networks in table (1) below:

Table 1 Performance of classification and Backpropagation networks

Attack Class	41 – Variable data set		
	Train(s)	Test (s)	Accuracy (%)
Normal	1.45	0.22	99.64
Prob	1.65	0.05	97.85
DOS	4.23	0.09	99.47
U2R	1.23	0.05	48.00
R2L	1.44	0.02	90.58

On the other hand, after the training and testing the *8* features show Performance of classification and Back propagation networks in Table (2) below:

Table 2 Performance of classification and Backpropagation networks

Attack Class	8 – Variable data set		
	Train(s)	Test (s)	Accuracy (%)
Normal	1.45	0.22	99.9216
Prob	1.65	0.05	99.3408
DOS	4.23	0.09	99.9341
U2R	1.23	0.05	100
R2L	1.44	0.02	97.4838

Attack Class	8 – Variable data set				
	Real Record	Neural Networks	Match Records	Miss Records	Accuracy
Normal	7656	7662	7650	12	99.9216
Prob	3944	3922	3918	4	99.3408
DOS	50040	50036	50007	29	99.9341
U2R	384	384	384	0	100
R2L	1391	1356	1356	0	97.4838
Unknown	0	55	0	55	Nan

And the Table (3) shows the amount of the data set use in the train and test:

Table 3 Show the amount the set

VIII. Conclusions

In this paper we have investigated new techniques for intrusion detection and performed data reduction and evaluated their performance on the DARPA benchmark intrusion data. Finally, we propose predicative model with high rate detect and base classifiers for intrusion detection. From the practical results, it is seen that by using the Neural Networks Normal, Probe and DOS could be detected with 99.9216% and 99.3408% and 99.9341% accuracy and U2R and

R2L with 100% and 97.4838% accuracies, respectively. Our future research will be directed towards developing more accurate base classifiers particularly for the

detection of R2L type of attacks. 'The percent accuracy of the Test' pert ' [99.3361], further more the percent of the 'Error ' [0.6639].

The table (4) below shows the number of the miss:

Table 4 Show the miss records

Attack Class	Miss Records
Normal	6
Prob	26
DOS	33
U2R	0
R2L	35
Unknown	0

The table (5) below shows the unknown:

Table 5 Show the unknown

Attack Class	Unknown
Normal	6
Prob	0
DOS	31
U2R	0
R2L	18

The table (6) below shows the percent of Alarm Rate:

Table 6 Show the percent of Alarm Rate

Attack Class	The percent of Alarm Rate
True Positive	99.8434
True negative	100
False negative	0.1566
False Positive	0

References:

[1]. Animesh Patcha *, Jung-Min Park." **Network anomaly detection with incomplete audit data** ".Bradley Department of Electrical and Computer Engineering, Elsevier: 2007

[2]. B. Yocom, R. Birdsall, D. Poletti-Metzel, Gigabit intrusion detection systems , http://www.nwfusion.com/reviews/2002/ 1104rev.html, 2002.

[3]. C. Sinclair, L. Pierce, and S. Matzner, "**An application of machine learning to network intrusion detection**," Proceedings of 15th Annual Computer Security Applications Conference (ACSAC '99), Phoenix, AZ, pp. 371-377, 1999.

[4]. Chebrolu S, Abraham A, Thomas JP." **Feature detection and ensemble design of intrusion detection systems**". Compute Secur; 24: 2005, pp.295–307.

[5]. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. Nature, 323:533–536, 1986.

[6]. H. Debar, M. Becker, and D. Siboni. "**A neural network component for an intrusion detection system**". In Proceedings of the 1992 IEEE Symposium On Research in Computer Security and Privacy, Oakland, CA, May 1992.

[7]. Http://kdd.ics.uci.edu/databases/kddcup99/task.htm.

[8]. K. Fox, R. Henning, J. Reed, and R. Simonian, "**A neural network approach towards intrusion detection**," Proceedings of 13th National Computer Security Conference, Baltimore, MD, pp. 125-134, 1990.

[9]. KDD 99 Task. Available at: http://kdd.ics.uci.edu/ databases/kddcup99/task.html, 1999.

[10]. Kristopher Kendall, "**A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems**," Masters Thesis, MIT, 1999

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

[12]. J. Cannady. "**Artificial Neural Networks for Misuse Detection**". In Proceedings of the 1998 National Information Systems Security Conference (NISSC'98), Arlington, VA, USA, October 5-8 1998 **[2].** MIT Lincoln Laboratory, http://www.ll.mit.edu.

[13]. MIT Lincoln Laboratory, http://www.ll.mit.edu.

[14]. 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.

[15]. Sergios Theodorios and Konstantinos Koutroumbas, **Pattern Recognition,** Cambridge: Academic Press, 1999.

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

Appendix:

The table (7) describes the 41 features of each connection record in the DARPA KDD cup 2000. The fields with blue color are features that have been considered in this research based on previous studies suggested by Chebrolu [4].

Table 7 features of KDD CUP 2000

Label	Network data Feature	Label	Network data Feature	Label	Network data Feature
A	duration	P	num_root	AE	srv_diff_host_rate
B	protocol-type	Q	num_file_creations	AF	dst_host_count
		R	num_shells	AG	dst_host_srv_count
D	flag	S	num_access_files	AH	dst_host_same_srv_rate
		T	num_outbound_cmds	AI	dst_host_diff_srv_rate
		U	is_host_login	AJ	dst_host_same_src_port_rate
G	land	V	is_guest_login	AK	dst_host_srv_diff_host_rate
H	wrong_fragment			AL	dst_host_serror_rate
I	urgent			AM	dst_host_srv_serror_rate
J	hot			AN	dst_host_rerror_rate

K	num_falied_logins	Z	srv_serror_rate	AO	dst_host_srv_rerror_rate
		AA	rerror_rate		
M	num_compromised				
N	root_shell	AC	same_srv_rate		
O	su_attempted	AD	diff_srv_rate		