International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
National Conference on Advances in Engineering and Technology
(AET- 29th March 2014)

RESEARCH ARTICLE OPEN ACCESS

A Hybrid Intrusion Detection System Using Genetic-Neural Network

Parveen Kumar, Nitin Gupta
Department of Computer Science and Engineering,
National Institute of Technology, Hamirpur
parveen.csed@gmail.com,nitin3041@gmail.com

ABSTRACT
Now days, with the dramatic growth in communication and computer networks, security have become a critical subject for computer systems. A good way to detect the illegal users is to monitoring these user's packets. Different algorithms, methods and applications are created and implemented to solve the problem of detecting the attacks in intrusion detection systems. In this paper, we present an intrusion detection model based on genetic algorithm and neural network. The key idea is to take advantage of classification abilities of genetic algorithm and neural network for intrusion detection system. The new model has ability to recognize an attack, to differentiate one attack from another i.e. classifying attack, and the most important, to detect new attacks with high detection rate and low false negative. This approach uses evolution theory to information evolution in order to filter the traffic data and thus reduce the complexity. To implement and measure the performance of this System. We used the KDD99 benchmark dataset and obtained reasonable detection rate.
Keywords: Genetic Algorithm, Intrusion Detection System, KDD Cup 1999 Dataset, Neural Network.

I. INTRODUCTION

With the rapid growth of the internet, computer attacks are increasing at a fast pace and can easily cause millions of dollar in damage to an organization. Detection of these attacks is an important issue of Computer security. Intrusion Detection Systems (IDS) technology is an effective approach in dealing with the problems of network security. In general, the techniques for Intrusion Detection (ID) fall into two major categories depending on the modeling methods used: misuse detection and anomaly detection. Misuse detection compares the usage patterns for knowing the techniques of compromising computer security. Although misuse detection is effective against known intrusion types; it cannot detect new attacks that were not predefined. Anomaly detection, on the other hand, approaches the problem by attempting to find deviations from the established patterns of usage. Anomaly detection may be able to detect new attacks. However, it may also cause a significant number of false alarms because the normal behavior varies widely and obtaining complete description of normal behavior is often difficult. Architecturally, an intrusion detection system can be categorized into three types**:** host based IDS, network based IDS and hybrid IDS [1] [2]. A host based intrusion detection system uses the audit trails of the operation system as a primary data source.

A network based intrusion detection system, on the other hand, uses network traffic information as its main data source. Hybrid intrusion detection system uses both methods [3]. However, most available commercial IDS's use only misuse detection because most developed anomaly detector still cannot overcome the limitations (high false positive detection errors, the difficulty of handling gradual misbehavior and expensive Computation [4]). This trend motivates many research efforts to build anomaly detectors for the purpose of ID [5]. The main problem is the difficulty of distinguishing between natural behavior and abnormal behavior in computer networks due to the significant overlap in monitoring data. This detection process generates false alarms resulting from the Intrusion Detection based on the Anomaly Intrusion Detection System. The use of Genetic algorithm might reduce the amount of false alarm, where Genetic algorithm is used to separate this overlap between normal and abnormal behavior in computer networks.

II. PREVIOUS WORK

In particular several Neural Networks based approaches were employed for Intrusion Detection. Several Genetic Algorithms (GAs) has been used for

detecting Intrusions of different kinds in different scenarios [6][7] [8] [9]. GAs used to select required features and to determine the optimal and minimal parameters of some core functions in which different AI methods were used to derive acquisition of rules [10] [11] [12]. In [13], authors presented an implementation of GA based approach to Network Intrusion Detection using GA and showed software implementation. The approach derived a set of classification rules and utilizes a support-confidence framework to judge fitness function. In [14], authors designed a GA based performance evaluation algorithm for network intrusion detection. The approach uses information theory for filtering the traffic data. In [15], authors used the BP network with GAs for enhancement of BP, they used some types of attack with some features of KDD data. A back-propagation Neural Network was used [16], authors used all features of KDD data, the classification rate for experiment result for normal traffic was 100%, known attacks were 80%, and for unknown attacks were 60%.

III GENETIC ALGORITHM

Genetic Algorithm is chosen to make this intrusion detection system. This section gives an overview of the algorithm and the system.

3.1. Genetic Algorithm Overview

A Genetic Algorithm (GA) is a programming technique that mimics biological evolution as a problem-solving strategy [17]. It is based on Darwinian's principle of evolution and survival of fittest to optimize a population of candidate solutions towards a predefined fitness [7].

GA uses an evolution and natural selection that uses a chromosome-like data structure and evolve the chromosomes using selection, recombination and mutation operators [7]. The process usually begins with randomly generated population of chromosomes, which represent all possible solution of a problem that are considered candidate solutions. From each chromosome different positions are encoded as bits, characters or numbers. These positions could be referred to as genes. An evaluation function is used to calculate the goodness of each chromosome according to the desired solution; this function is known as "Fitness Function". During the process of evaluation "Crossover" is used to simulate natural reproduction and "Mutation" is used to mutation of species [7]. For survival and combination the selection of chromosomes is biased towards the fittest chromosomes.

When we use GA for solving various problems three factors will have vital impact on the effectiveness of the algorithm and also of the applications [18].

They are:

i) The fitness function;
ii) The representation of individuals
iii) The GA parameters.

The determination of these factors often depends on applications and/or implementation.

3.2. Flowchart

Fig 1 shows the operations of a general genetic algorithm according to which GA is implemented into our system. Also all the three steps of generating new population from old population are depicted. The process of generating new population from old population includes selection, crossover, and mutation. If new population is not feasible then quit, otherwise again repeat the generation process.

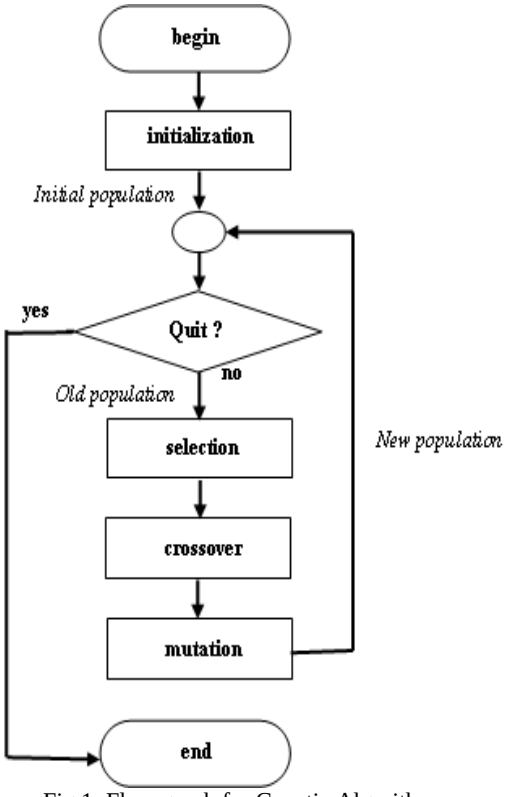

Fig 1: Flow graph for Genetic Algorithm

3.3. Steps of Precalculation System

This system can be divided into two main phases: the precalculation phase and the detection phase. Following are the major steps in precalculation phase, where a set of chromosome is created using training data. This chromosome set will be used in the next phase for the purpose of comparison.

Major Steps in Precalculation

Algorithm: Initialize chromosomes for comparison
Input: Network audit data (for training)
Output: A set of chromosomes

1. Range = 0.125
2. For each training data
3. If it has neighboring chromosome within Range
4. Merge it with the nearest chromosome
5. Else
6. Create new chromosome with it
7. End if
8. End for

IV. NEURAL NETWORK

Neural Networks (NNs) have attracted more attention compared to other techniques. That is mainly due to the strong discrimination and generalization abilities of Neural Networks that utilized for classification purposes [19]. Artificial Neural Network is a system simulation of the neurons in the human brain [20]. It is composed of a large number of highly interconnected processing elements (neurons) working with each other to solve specific problems. Each processing element is basically a summing element followed by an active 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 (weights) for solving a problem are found [21].

An increasing amount of research in the last few years has investigated the application of Neural Networks to intrusion detection. If properly designed and implemented, Neural Networks have the potential to address many of the problems encountered by rule-based approaches. Neural Networks were specifically proposed to learn the typical characteristics of system's users and identify statistically significant variations from their established behavior. In order to apply this approach to Intrusion Detection, I would have to introduce data representing attacks and non-attacks to the Neural Network to adjust automatically coefficients of this Network during the training phase. In other words, it will be necessary to collect data representing normal and abnormal behavior and train the Neural Network on those data. After training is accomplished, a certain number of performance tests with real network traffic and attacks should be conducted [22]. Instead of processing program instruction sequentially, Neural Network based models on simultaneously explorer several hypotheses make the use of several computational interconnected elements (neurons); this parallel processing may imply time savings in malicious traffic analysis .

V. EXPERIMENT DESIGN

The block diagram of the hybrid model is showed in the following Fig 2.

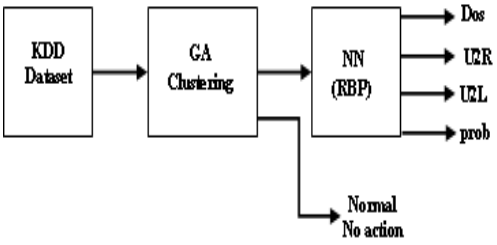

Fig 2: The block Diagram of Model

VI. KDD DATA SET

KDD 99 data set are used as the input vectors for training and validation of the tested neural network. It was created based on the DARPA intrusion detection evaluation program. MIT Lincoln Lab that participates in this program has set up simulation of typical LAN network in order to acquire raw TCP dump data. They simulated LAN operated as a normal environment, which was infected by various types of attacks. The raw data set was processed into connection records. For each connection, 41 various features were extracted. Each connection was labeled as normal or under specific type of attack.

There are 39 attacker types that could be classified into four main categories of attacks **:**

- DOS (Denial of Service): an attacker tries to prevent legitimate users from using a service. E.g. TCP SYN Flood, Smurf (391458 record).
- Probe: an attacker tries to find information about the target host. For example: scanning victims in order to get
- Knowledge about available services, using Operating System (4107 record).
- U2R (User to Root): an attacker has local account on victim's host and tries to gain the root privileges (52 records).
- R2L (Remote to Local): an attacker does not have local account on the victim host and try to obtain it (1124 records).

The KDD 99 intrusion detection benchmark consists of different components [22]:
kddcup.data;kddcup.data_10_percent;
kddcup.newtestdata_10_percent_unlabeled;
kddcup.testdata.unlabeled;
kddcup.testdata.unlabeled_10_percent; corrected.
We have used "kddcup.data_10_percent" as training dataset and "corrected" as testing dataset. In this case, the training set consists of 344,186 records

among which 95,140 are normal connection records, while the test set contains 159,696 records among which 59,260 are normal connection records. Table 1 shows the distribution of each intrusion type in the training and the test set.

VII. IMPLEMENTATION PROCEDURE

In the precalculation phase, this procedure will made 23 groups of chromosomes according to training data. There were 23 (22+1) groups for each of attack and normal types presented in training data. Number of chromosomes in each group is variable and depends on the number of data and relationship among data in that group. Total number of chromosomes in all groups were tried to keep in reasonable level to optimize time consumption in testing phase.

Table 1: Distribution of Intrusion Types in KDD Dataset

Dataset	Normal	Probe	Dos	u2r	r2l	Total
Train("kddcup.data_10_percent")	95140	4797	241458	67	2724	344186
Test("corrected")	59260	4166	79853	228	16189	159696

The anomaly detection is to recognize different authorized system users and identify intruders from that knowledge. Thus intruders can be recognized from the distortion of normal behavior. Because the Genetic Algorithm is classified normal from attack, the second stage of NN is used for classification of attacks type. Resilient Back-propagation networks (RBP) are used in this work. The number of hidden layers, and the number of nodes in the hidden layers, was also determined based on the process of trial and error. We choose several initial values for the network weight and biases. Generally these chosen to be small random values .The Neural Network was trained with the training data which contains only attack records as shown in Table 1. When the generated output result doesn't satisfy the target output result, the error from the distortion of target output was adjusted. Retrain or stop training the network depending on this error value. Once the training was over, the weight value is stored to be used in recall stage. The result of the training stage of different network architectures with different training algorithms and different activation functions is shown in the given Table 2.

Table 2: Test performance of different Neural Network Training functions

Function	No of Epochs	Accuracy (%)
Gradient Descent	2730	61.70
Gradient descent with moment	2730	51.60
Resilient back propagation	52	98.04
Scaled conjugate gradient	274	80.87
BFGS quasi-Newton method	282	75.67
One step secant method	498	89.60
Levenberg-marquardt	29	79.34

As seen from above Table 2 the best training algorithm is Resilient back propagation which takes less time, low no. of epoch, and high accuracy, therefore We used it in this paper. The Architecture based on this program used one hidden layer, consisting of 16 neurons and 4 neurons in the output layer, the desired mean square error is 0.00011 and the No. of Epoch is 52, the result of training is illustrated in Table 3.

Table 3: The training experiment of Resilient back propagation

	Input	Output	Accuracy
Dos	28580	28580	100%
U2R	11	11	100%
U2L	528	528	100%
Prob	2662	2662	100%
MSE		0.00011	
Time		00:00:54	
Epoch		52	

VIII. TEST AND RESULTS

The model is designed to provide output values between 0.0 and 1.0 in the output nodes. The first stage of the model is Genetic Algorithm for clustering, the classification rate is 99.99% which means that the false negative rate is 0.01% and the false positive rate is 0.01% as mentioned previously the manner of calculation them, is very low according to the previous researches. Genetic algorithm separates the normal records from attack records, then the RBP stage is the classification of attack to four types. During the testing phase, the accuracy classification of each attack types was calculated, classification time of two different inputs of datasets, the result is shown in Table 4.

Table 4: The Result of the Testing Phase

Attack name	Input 1	Output	Accuracy	Input 2	Output	Accuracy
Dos	28580	28581	99.9%	25687	25687	**100%**
U2R	11	11	100%	8	8	**100%**
U2L	528	528	100%	10	4	**40%**
Prob	2662	2662	100%	1330	1332	**99.8%**
Unknown	34	32	94.1%	228	332	**68.6%**
Time(sec)	5.8292			4.6766		

IX. CONCLUSION

In this paper, we present and implemented an Intrusion Detection System by applying genetic algorithm with Neural Network to efficiently detect various types of

network intrusions. To implement and measure the performance of our system, we used the standard KDD99 benchmark dataset and obtained reasonable detection rate. The second stage of the model is Neural Network. After many experiment on the Neural Network using different training algorithms and object functions, We observed that Resilient back propagation with sigmoid function was the best one for classification therefore We used it in this work. And trail many architectures with one hidden layer and two hidden layers with different number of neurons to obtain the best performance of the Neural Network.

REFERENCES

[1] J., Muna. M. and Mehrotra M., "Intrusion Detection System : A design perspective", *Proceeding of 2rd International Conference On Data Management, IMT Ghaziabad, India.,2009,265-372.*

[2] M. Panda, and M. Patra, "Building an efficient network intrusion detection model using Self Organizing Maps", *Proceeding of world academy of science, engineering and technology, 38, 2009, 22-29.*

[3] M. Khattab Ali, W. Venus, and M. Suleiman Al Rababaa, "The Affect of Fuzzification on Neural Networks Intrusion Detection System", *IEEE computer society,2009, 1236-1241.*

[4] B. Mykerjee, L. Heberlein T., and K. Levitt N., "Network Intrusion Detection", *IEEE Networks, 8(3), 1994, 14-26.*

[5] W. Jung K., "Integration Artificial Immune Algorithms for Intrusion Detection", *dissertation in University of London, 2002, 1-5.*

[6] A. Chittur, "Model Generation for an Intrusion Detection System Using Genetic Algorithms", *Technical Report, Ossining, New York, 2001.*

[7] W. Li, "Using Genetic Algorithm for Network Intrusion Detection", *http://www.security.cse.msstate.edu, Department of Computer Science and Engineering, Mississippi State University, USA, 2004.*

[8] W. Lu, I. Traore, "Detecting New Forms of Network Intrusion Using Genetic Programming", *Computational Intelligence, 20(3), Blackwell Publishing, Malden, 2004, 475-494.*

[9] M. M. Pillai, J. H. P. Eloff, H. S. Venter, "An Approach to Implement a Network Intrusion Detection System using Genetic Algorithms", *Proceedings of SAICSIT, 2004, 221-228.*

[10] S. M. Bridges, R. B. Vaughn, "Fuzzy Data Mining And Genetic Algorithms Applied To Intrusion Detection*", Proceedings of 12th Annual Canadian Information Technology Security Symposium, 2004, 109-122.*

[11] J. Gomez, D. Dasgupta, "Evolving Fuzzy Classifiers for Intrusion Detection", *Proceedings of the IEEE, 16(6), 2002, 1462-1475.*

[12] R. H. Gong, M. Zulkernine, P. Abolmaesumi, "A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection", *IEEE, 2005, 246-253.*

[13] B. Abdullah, I. Abd-alghafar, Gouda I. Salama, A. Abd-alhafez, "Performance Evaluation of a Genetic Algorithm Based Approach to Network Intrusion Detection System", *Proceeding of 13th International Conference on AEROSPACE SCIENCES & AVIATION TECHNOLOGY, 2009, 1-17.*

[14] M. Vallipuram and B. Robert, "An Intelligent Intrusion Detection System based on Neural Network", *Proceeding of International Conference on Applied Computing, 2004, 356-362.*

[15] R. H. Gong, M. Zulkernine, P. Abolmaesumi, "A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection", *Proceeding of 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005, 246-253.*

[16] M. Al-Subaie, "The power of sequential learning in anomaly intrusion detection", *master degree thesis, Queen University, Canada, 2006.*

[17] P. Kukielka and Z. Kotulski, "Analysis of different architectures of neural networks for application in intrusion detection systems", *proceeding of the international conference on computer science and information technology, 2008, 807-811.*

[18] M. Moradi and M. Zulkernine, "A Neural Network based system for intrusion detection and classification of attacks", *submitted at Queen University, Canada, 2004, 148-154.*

[19] D. Novikov, V. Roman Yampolskiy, and L. Reznik, "Artificial Intelligence Approaches For Intrusion Detection", *Proceeding of Systems, Applications and Technology Conference, IEEE Long Island, 2006, 1-8.*

[20] S. Lília de Sá, C. Adriana Ferrari dos Santos, S. Demisio da Silva, and A. Montes, "A Neural Network Application for Attack Detection in Computer Networks", *Proceeding of IEEE joint conference on Neural Networks, 2, 2004, 1569-1574.*

[21] P. Kukiełka and Z. Kotulski, "Analysis of Different Architectures of Neural Networks for Application in Intrusion Detection Systems", *Proceedings of the International Multi conference on Computer Science and Information Technology, IEEE, 2008, 807– 811.*

[22] KDD Cup 1999: Data; http://www.kdd.org/kddcup/index.php?section=1999&method=data