THE ANNALS OF "DUNĂREA DE JOS" UNIVERSITY OF GALATI
FASCICLE III, 2014, VOL. 37, NO. 1, ISSN 2344-4738, ISSN-L 1221-454X
ELECTROTECHNICS, ELECTRONICS, AUTOMATIC CONTROL, INFORMATICS

STUDY ON INTRUSION DETECTION SYSTEMS USING GENETIC ALGORITHM

Marjan Kuchaki Rafsanjani , Milad Riyahi

Department of Computer Science, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran
kuchaki@uk.ac.ir, milad.rcd65@gmail.com

Abstract: With the extension in computer networks and appearing new attacks, seems that security is more necessary than before. Intrusion Detection System (IDS) is one of the most important methods to develop security in computer networks. There are different methods for IDS improvement. Machine learning is one of these methods and an approach for improving IDS with machine learning using Genetic Algorithm (GA).

Keywords: Computer networks, Intrusion Detection System, Machine learning, Genetic Algorithm

1. INTRODUCTION

In recent years, computer networks, like Internet, are spreading at amazing rate, not only in size of networks, but also in different services which provided in mobility of users (Bankovic, et al., 2007). New services and users mobility have a lot of performances and benefits, but this service causes new problems and new threats. This spreading leads to the huge amount of data which needs to protect against malicious and many different and complex kind of attacks which raid to network. In this situation, the need for protection of network is increased and it is a very important issue in computer networks. When an attack occurred, the network needs for an immediate and correct reaction, and this reaction is answered by the security system to attack or intruder (Li, 2004). There are different kind of approaches to protect networks, like anti-viruses, firewall, password protection and so on, but with these techniques, having a completely secure network is impossible, therefore, need to a powerful security system is very serious and necessary (Bankovic, et al., 2007). A method for security of networks is IDS which works based on normal or abnormal behavior (Chittur, 2001). Many different of soft-computing like fuzzy logics and neural networks (Pan, et al., 2003; Yao, et al., 2005) used for improving IDS. One of these practical approaches is Genetic Algorithm (GA).

The rest of the paper is organized as follows. In section 2, Intrusion Detection System (IDS) is described. In section 3, a data set which used for IDS is introduced. In section 4, Genetic algorithm (GA) will be explained and after that we will discuss on approaches which combine GA and IDS in section 5. In section 6, the approaches will be compared and finally, we will conclude the paper in section 7.

2. INTRUSION DETECTION SYSTEM

IDS is an alternative solution for the security of network problems. This approach behaves based on user behavior. Behavior of users can be defined according to a set of properties which exist in a connection, between client and server. When an internal or an external client connects to network, IDS checks connection properties and decides that whether this connection is normal or an attack (Chittur, 2001). Using IDS has following important benefits (Sinclair, et al., 1999; Lee, et al., 2001):

This paper was recommended for publication by Viorel Nicolau

- IDS doesn't need to a primary attack for diagnosis future attacks because there is completely difference between normal and malicious behaviors.

- IDS doesn't need to update because any attack will not main change throughout the system's lifetime due to inherent, attacks are different from normal connections.

There are two main categories for IDS: Anomaly IDS and Misuse IDS. Anomaly is a kind of IDS where is based on normal behaviors. In fact, in this category of IDS, properties of some normal connections are saved and other connection properties are compared with them. In this comparison, if matching occurs, the connection is normal and otherwise the connection is malicious. In misuse IDS, unlike anomaly, properties of malicious connections are saved and other connection properties are compared with them. If matching occurs, the connection is malicious and if mismatching occurs, the connection is normal (Chittur, 2001).

3. DATA SET

In IDS, properties of normal or malicious connection are saved and then each connection property is compared with that, therefore, IDS needs properties which are important in a connection. This set of properties is called DATA SET. KDD99CUP is one of these data sets which contain 41 attributes, like the duration of a connection, which are important in a connection (KDD Cup, 1999). Each property has a value in a connection and IDS can recognize malicious connection according to this value.

4. GENETIC ALGORITHM

Improvement in IDS is an important goal. For achieving to this important goal, different machine learning techniques like C4.5, MLP, and SVM have been used and each of them tested for building more powerful IDS (Bankovic, et al., 2007). One of these approaches is GA. This algorithm presented by John Holand in 1975 (Holland, 1975). GA works based on Darwinian evolution theorem (Chittur, 2001). In fact, GA is a random search algorithm for finding global optimum based on natural selection and also GA can solve nonlinear problems (Yang, et al., 2013). In figure 1, simple structure of the GA is shown. GA starts with an initial population of individuals which each of them is a potential solution. GA attempts to improve population, according to a fitness function at each iteration. In the next section the overall structure of four heuristic approaches that developed in the last decade for IDS improvement using GA, are described.

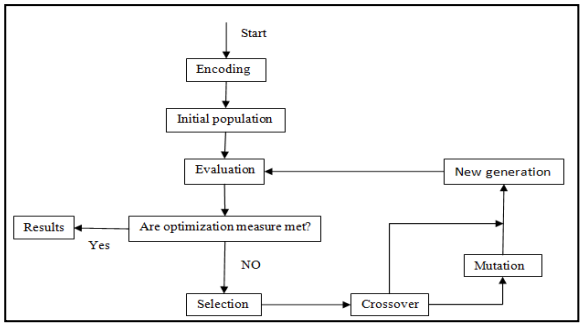

Fig.1. Structure of simple Genetic Algorithm.

5. REVIEWING APPROACHES

In this section, 4 approaches which using GA for improving IDS are presented in separated sections.

5.1. Chittur's Approach

This approach is presented by Chittur in 2001 (Chittur, 2001). The main goal of this paper was to test whether GA is a feasible solution for IDS or not. Chittur represented each connection as a chromosome and generate rules with GA for IDS knowledge base and also generate some coefficient whit it. Chittur used a decision tree model where each node holds a random coefficient. These coefficients which generated with GA, are multiplied by data to find whether a certain connection is an attack or not. Chittur used KDD99CUP, so each connection had 41 attributes. These attributes are multiplied with coefficients and its results was a weight for each connection. Chittur established an arbitrary threshold. A connection was classified as malicious if weight of connection exceeded that threshold. Chittur defined fitness function based on how many attacks detected and how many normal connection classified as an attack. The formula which Chittur used for fitness function for an individual δ_i is:

$$(1)\ F(\delta_i)=\frac{\alpha}{A}-\frac{\beta}{B}$$

Where α is the number of detected attacks and A is the number of total attacks, β is the number of false positive and B is the number of total correct connections. A connection which is correct, but classify as attack connection, are called false positive. Firstly, Chittur executed algorithm for twice on 10% of data and after that performed it on all of the data.

5.2. Li's Approach

This approach is presented by Li (Li, 2004). Li changed the rules of the network to chromosomes

and used special structure: IF {condition} THEN {act}. This structure means that if the condition occurred, do special act. This act is determined by a security system of network. The condition refers to matches between current connection and rules in the database of IDS. According to match or mismatch, attack or normal was defined, and if the connection was attacked, security system defined a reaction. For example:

IF {the connection has following information:

source IP address: 124.12.5.15; destination IP address: 130.18.206.55; destination port number: 21;

connection time: 10.1 seconds}

THEN {stop the connection}

This rule says if special conditions like special source IP address or special destination IP address occurred, stop the connection. Li used GA to generate rules for normal connections and then compare each connection with them for recognizing malicious connections. For fitness function, Li defined some variables. Firstly, Li defined Outcome which was calculated by multiplying weight of field of connection with matching between pre-data and field of connection. The outcome was calculated by following formula:

$$(2)\ \text{Outcom} = \sum_{i=1} (\text{Matched} * \text{weight}_i)$$

In this formula Matched value is either 1 or 0. Li considered a weight for each field of a connection according to the importance of the field. For example, destination IP address was most important field in a connection. Li also defined another variable, Suspicious level which it was a threshold level that indicated which two connections could consider "matched". Li calculated difference between Outcome and Suspicious level with formula (3):

$$(3)\ \Delta = |\text{Outcome} - \text{suspicious level}|$$

If a mismatch was occurring, Li calculated it with Penalty and used formula (4):

$$(4)\ \text{Penalty} = \left(\frac{\Delta * \text{ranking}}{100} \right)$$

Ranking in this formula is whether or not an intrusion is easy to find. Li defined fitness function according to Penalty with the following equation:

$$(5)\ \text{Fitness} = 1 - \text{penalty}$$

Unfortunately, Li didn't simulate this approach on a network, but this algorithm changed how to use GA for IDS and in other approaches, researchers use a definition of rules that Li presented.

5.3. Bankovic's Approach

This algorithm was presented in 2007 (Bankovic, et al., 2007). According to KDD99CUP, each connection has 41 attributes. These attributes lead to huge amounts of data and cause difficulty in researches. BANKOVIC' and her colleagues decided that for reducing in the amount of data reduce attributes. PCA was used for this purpose. They extracted some attributes based on kinds of attacks. They thought that the fitness function which presented by Chittur showed only total number of intrusions, so in this research, researches introduced another fitness function:

$$(6)\ \text{Support} = |A \text{ and } B| \,/\, |N|$$

$$(7)\ \text{Confidence} = |A \text{ and } B| \,/\, |A|$$

$$(8)\ \text{Fitness} = (w_1 * \text{Support}) + (w_2 * \text{Confidence})$$

In the above formulas, N is the total number of network connection, |A| is a symbol for the number of connections that matches with condition A and also |A and B| is a symbol for the number of connections that matches the rule "IF A THEN B". The weights w_1 and w_2 use for balancing in two terms. In this approach, they used PCA for reducing attributes. After running PCA on 41 attributes, 3 of them are selected. These attributes were, *Src_bytes* and *Duration*, which each of them was a gene on a chromosome. They generated an initial population, then performed GA on this population. After performing GA, they used 10 chromosomes with maximum fitness for intrusion detection. Then they attacked to network with 3 different following attacks: *Neptune*, *Smurf* and *Port Sweep*. Also, they run this algorithm for two different fitness functions and also they run their algorithm on training and test data.

5.4. Sadiq Ali Khan's Approach

This approach was presented in 2011 (Sadiq Ali Khan, 2011). Like previous approach, Sadiq Ali Khan used PCA for reducing attributes. After running PCA, the following attributes are selected:

Service, *flag, land, logged_in , root_shell , su_attempted , is_hot_login , is_guest_login*

These attributes were as genes of a chromosome. Also the author for defining rules used IF (condition) THEN (act) structure. Sadiq Ali Khan used following fitness function:

(9) Fitness = $\sum_{i=1}$ number of matches * weight of field

Like previous approaches, this approach runs this algorithm on both training and test data.

6. COMPARING APPROACHES

In table 1, each of approaches results is shown. As mentioned, unfortunately, Li didn't simulate his approach so in this table, 3 approaches are compared.

Table 1 Comparing results

Traning				
Methods	Normal	Attack	False positive (%)	Detection rate (%)
Chittur	97276	369743	0.6877	97.4694
Bankovic	839	137	0	92.74
Sadiq Ali Khan	5000	5139	10.8	94.64
Test				
Methods	Normal	Attack	False positive (%)	Detection rate (%)
Chittur	927779	3925650	0.306	97.7601
Bankovic	743	234	1.62	94.87
Sadiq Ali Khan	5040	4958	6.55	94.19

As shown in table 1, in all of the approaches, presented algorithms were performed for twice. For first time algorithms run on training data and after that run on the test set. Chittur approach has highest Detection rate and least False positive in both training and test. In fact, this approach has most accuracy for detecting intrusions. But running speed in this approach is less than others because in other approaches, researchers reduce attributes and it caused more speed. As said for Bankovic's approach, they used 3 kinds of attacks. They used 74 Neptune attacks, 24 Smurf attacks, and 39 Port Sweep attacks in training, and also, 87 Neptune attacks, 107 Smurf attacks, and 40 Port Sweep attacks for testing their algorithm. Also, as said, they introduced another fitness function which after running the algorithm with this fitness function only 87.6% of attacks detected correctly. This detection rate shows that Chittur fitness function is more effective. The results also show that there is no great distance between detection rate in each run on test and training data and it means that number of connections is not very important issue for accuracy.

7. CONCLUSION

In this paper, 4 approaches of IDS that using GA are investigated. Results show that GA is an impressive approach for IDS. High detection rate and low false positive shows that using GA, on both of offline or training and online or test data, is effective. These results show that trade-off between time and accuracy is really needed. Using what approach is problem oriented and completely relate to the network. If time is important so attributes could be reduced and if the accuracy is necessary all attributes could be used.

8. REFERENCES

Bankovic', Z.**,** D. Stepanovic, S. Bojanic and O. Nieto-Taladriz (2007). Improving network security using genetic algorithm approach**.** *Computer and Electrical Engineering*, **Vol. 33**, pp. 438-451.

Chittur, A. (2001). Model generation for an intrusion detection system using genetic algorithms. *Ossining High SchoolHonors Thesis*, Ossining NY.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. *University of Michigan Press, Ann Arbor*.

Lee, W., S. Stolfo, P. Chan, E. Eskin, W. Fan, M. Miller, S. Hershkop and J. Zhang (2001). Real time data mining-based intrusion detection. *Proceedings of the DARPA Information Survivability Conference & Exposition II (DISCEX '01)*, **Vol. 1**, Anaheim, CA, pp. 89-100.

Li, W. (2004). Using genetic algorithm for network intrusion detection. *Proceedings of the United States Department of Energy Cyber Security Group 2004 Training Conference*.

Pan, Z., S. Chen, G. Hu and D. Zhang (2003). Hybrid neural network and C4.5 for misuse detection. *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, **Vol. 4**, pp. 2463–2467.

Sadiq Ali Khan, M. (2011). Rule based network intrusion detection using genetic algorithm. *International Journal of Computer Applications*, **Vol. 18, No. 8**. pp. 26-29.

Sinclair, C., L. Pierce and S. Matzner (1999). An application of machine learning to network intrusion detection. *Proceedings of the 15th Annual Computer Security Applications Conference*, pp. 371-377.

Yang, H., J. YI, J. Zhao and Z. Dong (2013). Extreme learning machine based genetic algorithm and its application in power system economic dispatch. *Neurocomputing*. **Vol. 102**, pp. 154-162.

Yao, J., S. Zhao and L. Saxton (2005). A study on fuzzy intrusion detection. *Proceedings of SPIE, Data Mining, Intrusion Detection, Information Assurance, And Data Networks Security, Orlando, Florida, USA*, pp. 23–30.