Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 5, May 2014, pg.1231 – 1237

RESEARCH ARTICLE

Intrusion Detection using Fuzzy Data Mining

Sandeep Dhopte[1], Prof. Manoj Chaudhari[2]

[1]Computer Engineering, PBCE, RTM Nagpur University, India

[2]Computer Engineering, PBCE, RTM Nagpur University, India

[1] sandy_22ce@yahoo.com

Abstract— With the rapid expansion of computer networks during the past few years, security has become a crucial issue for modern computer systems. A good way to detect illegitimate use is through monitoring unusual user activity. The solution is an Intrusion Detection System (IDS) which is used to identify attacks and to react by generating an alert or blocking the unwanted data. For IDS, use of genetic algorithm gives huge number of rules which are required for anomaly intrusion detection. These rules will work with high-quality accuracy for detecting the Denial of Service and Probe type of attacks connections and with appreciable accuracy for identifying the U2R and R2L connections. After getting huge rules we apply fuzzy data mining techniques to security system and build a fuzzy data mining based intrusion detection model. These findings from this experiment have given promising results towards applying GA and Fuzzy data mining for Network Intrusion Detection. Performance of the proposed system will be measured using the standard KDD 99 data set.

Keywords- IDS, Genetic Algorithm, KDDCUP dataset, rule set, Data Mining, Fuzzy Logic

I. INTRODUCTION

In recent years, the Internet and local area networks are developing and expanding at a very high speed. While we are benefiting from the convenience that the new technology has brought us, computer systems are facing increased number of security threats that originate externally or internally. As malicious intrusions into computer systems have become a growing problem, the need for accurately detecting these intrusions has risen. Despite numerous technological innovations for information assurance, it is still very difficult to protect computer systems. Therefore, intrusion detection is becoming an increasingly important technology that monitors network traffic and identifies, preferably in real time, unauthorized use, misuse, and abuse of computer systems, such as anomalous network behaviors, unauthorized network access, and malicious attacks to computer systems [1].

When an intruder attempts to break into an information system or performs an action not legally allowed, we refer to this activity as an intrusion. Intruders can be divided into two groups, external and internal. The former refers to those who do not have authorized access to the system and who attack by using various penetration techniques. The latter refers to those with access permission who wish to perform unauthorized

activities. Intrusion techniques may include exploiting software bugs and system misconfigurations, password cracking, sniffing unsecured traffic, or exploiting the design flaw of specific protocols. An Intrusion Detection System is a system for detecting intrusions and reporting them accurately to the proper authority [5].

There are two generally accepted categories of intrusion detection techniques: misuse detection and anomaly detection. In misuse detection, the IDS analyzes the information it gathered and compares it to large database of attack signatures, and anomaly detection systems identify deviations from normal behaviors of network's traffic and alarm for potential unknown attacks [1]. Hybrid detection systems combine both the misuse and anomaly detection systems. The categorization IDS's and also be done with respect to the location of intrusion. The activities with a particular host can be monitored by a host based IDS, monitoring the network traffic is done by a network-based IDS. The host activities like system calls, application logs, password files, capability databases can be tested for intrusion detection by a host based IDS. The network traffic and individual packets for mischievous traffic is tested by a network based IDS [7], [8].

II. GENETIC ALGORITHM (GA)

GA evolves a population of initial individuals to a population of high quality individuals, where each individual represents a solution of the problem to be solved. Each individual is called chromosome, and is composed of a predetermined number of genes [3]. The quality of each rule is measured by a fitness function as the quantitative representation of each rule's adaptation to a certain environment. The procedure starts from an initial population of randomly generated individuals [13]. Then the population is evolved for a number of generations while gradually improving the qualities of the individuals in the sense of increasing the fitness value as the measure of quality [11]. During each generation, three basic genetic operators are sequentially applied to each individual with certain probabilities, i.e. selection, crossover and mutation. The algorithm flow is presented in Fig.1 [2].

After initial population generation, if the number of individual equals to the required number then perform genetic operation on each individual, else generate more individuals. Perform the same operation any number of times (variable number) and then stop.

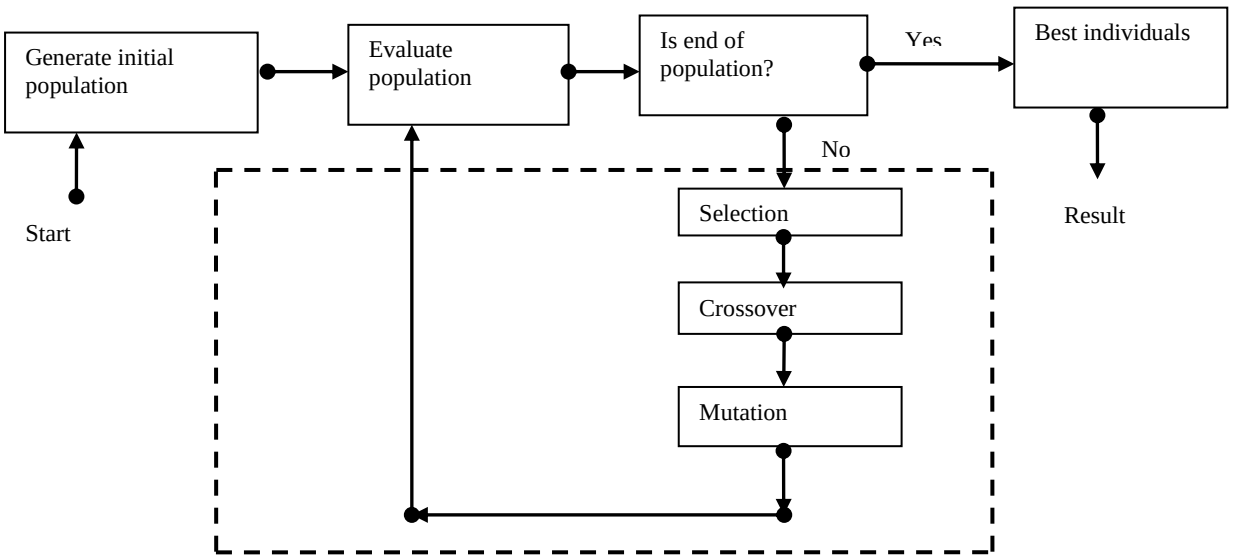

Fig. 1: Flowchart of GA system [2]

Three kinds of genetic operators, i.e., selection, mutation, and crossover are [6]:

A. Selection

Reproduction (or selection) is an operator that makes more copies of better strings in a new population. Reproduction is usually the first operator applied on a population [12]. Reproduction selects good strings in a population and forms a mating pool. This is one of the reasons for the reproduction operation to be sometimes known as the selection operator. Thus, in reproduction operation the process of natural selection causes those individuals that encode successful structures to produce copies more frequently.

B. Crossover

A crossover operator is used to recombine two strings to get a better string. In crossover operation, recombination process creates different individuals in the successive generations by combining material from two individuals of the previous generation. In reproduction, good strings in a population are probabilistically assigned a larger number of copies and a mating pool is formed. It is important to note that no new strings are formed in the reproduction phase. In the crossover operator, new strings are created by exchanging information among strings of the mating pool [10].

The two strings participating in the crossover operation are known as parent strings and the resulting strings are known as children strings. It is intuitive from this construction that good sub-strings from parent strings can be combined to form a better child string, if an appropriate site is chosen. With a random site, the children strings produced may or may not have a combination of good sub-strings matter of serious concern, because if good strings are created by crossover, there will be more copies of them in the next mating pool generated by crossover. So it is clear that the effect of crossover may be beneficial. Thus, in order to preserve some of the good strings that are already present in the mating pool, all strings in the mating pool are not used in crossover.

C. Mutation

Mutation in a way is the process of randomly disturbing genetic information. They operate at the bit level; when the bits are being copied from the current string to the new string, there is a probability that each bit may become mutated. This probability is usually a quite small value, called as *mutation probability*. This helps in introducing a bit of diversity to the population by scattering the occasional points. This random scattering would result in better optima, or even modify a part of genetic code that will be beneficial in later operations. On the other hand, it might produce a weak individual that will never be selected for further operations.

III. FUZZY DATA MINING ALGORITHM

Fuzzy Association Rules Agrawal and Srikant [2] defined an association rule using the following notation: n transactions, {T1 , T2 , …, Tn} of m items, { i1, i2, ..., im} in the set of all items, I. Each of the transactions Tj ($1 \leq j \leq n$) in the database D represents an association of items ($Tj \subseteq I$). An itemset is defined as a non-empty subset of I. An association rule can be represented as: $X \rightarrow Y$, c, s, where $X \subseteq I$, $Y \subset I$, and $X \cap Y = \emptyset$. In this association rule, s is called support and c is confidence of the association rule. The support is the percentage of the transactions in which both X and Y appear in the same transaction and the confidence is the ratio of the number of transactions that contain both X and Y to the number of transactions that contain only X. Originally, Agrawal and Srikant [2] implemented the Apriori algorithm to mine single-dimensional Boolean association rules from transactional databases [5]. However, in the intrusion detection field we need to mine quantitative attributes.

The basic Apriori algorithm [2] finds frequent itemsets for Boolean association rules, receiving as input a database T of transactions and the minimum support for the rules. It uses the Apriori property: if an itemset I is not frequent, the itemset $I \cup A$ (A is any other item) is also not frequent; i.e. "all nonempty subsets of a frequent itemset must also be frequent" [5, pp. 231].

IV. PROPOSED ALGORITHM FOR IDS

The key of data mining of intrusion detection lies in how to effectively distinguish normal behaviors and abnormal behaviors from plenty of initial data attributes and how to automatically and effectively generate intrusion rules after collecting the initial data of network. The preceding association data mining algorithm can be used for intrusion detection. We can carry out feature pattern extraction of user or system behavior through the above data mining algorithms. The proposed model of Intrusion Detection System (IDS) is depicted in fig.2 which consists of the following two phases:

Pre-processing: In this phase, extract the most suitable features from KDD data set, convert the symbolic features into numerical ones and normalize the data set.

Learning Phase: In this phase, genetic algorithm is used to generate/produces more number of rules, which are required for better testing. Here, proposed system is trained using the newly formed training data set.

A. *Pre-processing*

Rules are required for detecting intrusions so more and more rules required for better performance of the system. For these, rules from KDD are used. As KDD data set contains 41 features, use of all features are not possible, so here we are using 10 features of KDD Data set. These features are duration, protocol type, count, flag, service, land, src_bytes, dst_bytes, urgent, and wrong-fragment. Inputs of some important features of this sorted dataset are given to the pre-processor. Steps for pre-processing of features are as follows:

Algorithm: Feature extraction
Step 1: Select some features from KDD dataset
Step 2: Convert symbolic features into numeric ones
Step 3: Rules in numeric form given to Learning phase

B. *Genetic Algorithm (GA)*

Rules from pre-processing are in numeric form. There is no way to clearly identify whether a network connection is normal or anomalous just using one rule. Multiple rules are needed to identify an unrelated anomaly, which means that several good rules are more effective than a single best rule. Here, genetics is used to increase the number of rules in the rule pool. By using genetic algorithm, more number of rules will be generated. These rules are required for misuse detection and anomaly detection. So common algorithm we can apply for both types of rules generation.

Following algorithm explains about the genetic operators which are used:

Algorithm: Genetic operators
Step 1: Take a input from the pre-processing unit
Step 2: Select random two parents for crossover
Step 3: One point crossover is used here
Step 4: Apply mutation operator on selected parents
Step 5: Check whether newly generated rules are already in database
Step 6: If rules are new then check the fitness value of each rule
Step 7: For fitness function,
Calculate the number of connections Ntc correctly detected by rule r
Calculate the number of connections in the training data Nt
Calculate the number of normal connections Nni incorrectly detected by rule r
Calculate the number of normal connections in the training data Nn
Calculate Fitness value of new rule

$$fitness_r = \frac{Nt_c}{Nt} - \frac{Nn_i}{Nn}$$

Step 8: If fitness is greater or equal to some threshold value then,
Add newly generated rule to rule pool

In each generation, apply crossover and mutation to increase the number of rules. For Crossover, first, parents are determined by selecting two individuals from the rule pool. Here, a single point crossover is used to reproduce more rules. In a single point crossover, exchange of genes (features value) between two rules with respect to some point is carried out. Mutation is applied on each rule to produce the new rule. This proposed system is mainly for producing more rules and which are efficient one. For detection of intrusion we required fittest rules in huge quantity.

C. *Fuzzy Logic*

After applying a genetic algorithm on normal and intrusion rule pool, all possible combinations of rules will reproduce. On a large dataset, now apply fuzzy logic to avoid the sharp boundary problem. In this module, types of attributes i.e. discrete and continuous are used. The following algorithm shows fuzzy logic implementation for the rule pool.

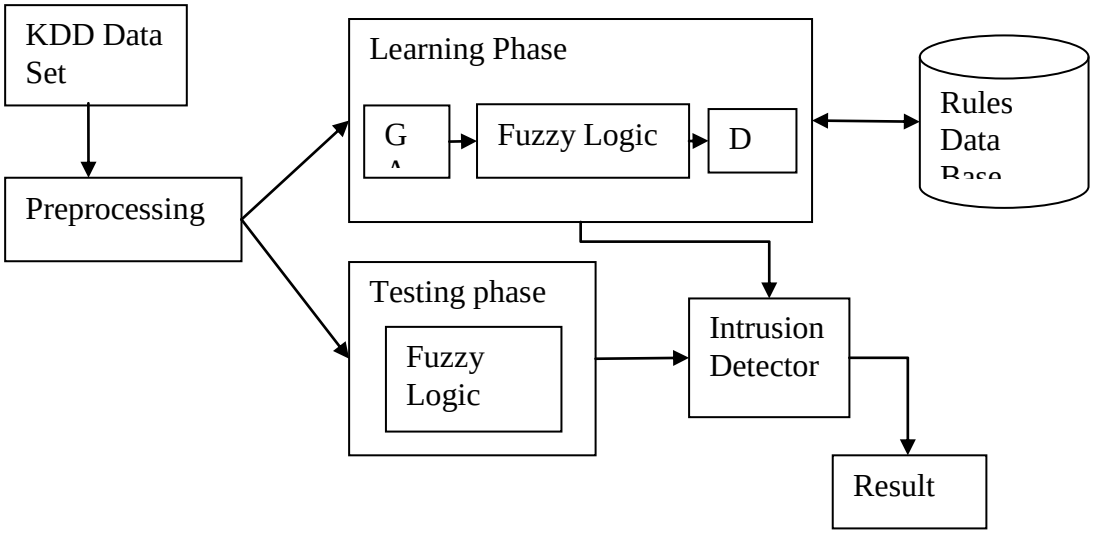

Fig.2: Proposed System for Intrusion Detection

Algorithm: Fuzzy Algorithm.
Step 1: Select features from rule pool
Step 2: Check for missing entry for all records
Step 3: Select record from the rule pool
Step 4: Process all selected attribute
Step5: Divide each continuous attribute into LOW, MEDIUM and HIGH
Step6: Set fuzzy membership value for each continuous attribute

$$\alpha + \gamma = 2\beta$$

Step7: Calculate fuzzy membership value for each attribute
Step 8: Store all fuzzy rules in fuzzy rule pool
Step 9: Repeat above steps for all rules
The parameters α, β, and γ in a fuzzy membership function for attribute A_i is set as follows:
β = average value of attribute A_i in the database
γ = the largest value of attribute A_i in the database

$$\alpha + \gamma = 2\beta$$

D. *Algorithm for Association-Rule Mining (ARM)*

After fuzzy implementation, the fuzzy rule pool will be generated and this rule pool is given as an input to association rule mining. For rule generation, antecedent part is generated by using apriori algorithm and for consequent; classification method is used in which the whole KDD dataset is distributed into two classes i.e. normal and attack class on the basis of labels provided in the dataset. Following algorithm is used for finding the frequent itemsets from the dataset i.e. Apriori algorithm:

Algorithm: Apriori algorithm for finding frequent itemsets.
Step 1: Scan each record of the fuzzy rule pool.
Step 2: Find frequent itemset L_k from C_k of all candidate itemsets

Scan D and count each itemset in C_k,
If it is greater than minimum support, then it is frequent
Step 3: Form C_{k+1} from L_k; k = k + 1

Join L_{k-1} itemset with itself to get the new candidate itemsets,

If found a non-frequent subset then remove that subset.
Step 4: Store frequent itemset in the rule pool
Step 5: Repeat step 5 and step 9 until C_k is empty

In the above algorithm, for candidate itemset generation scan the dataset i.e rule pool. In the first candidate itemset, all attributes will appear but for the second, third and further itemsets generate a different combination of all items. After each candidate itemset, find support for each attribute and compare with minimum support, if greater than minimum support then keeps that attribute (or combination of attributes). Large itemset contains

frequent items which is stored in rule pool. At the end of this algorithm, rule pool contains rules which are used for testing purpose of the system.

V. OBSERVATIONS

Our aim for using genetic algorithm is to produce huge number of rules for detection. For anomaly detection required more normal rules and for misuse detection we required best quality rules. GA is best for producing more rules and increasing the database. In crossover, two new rules produce and in mutation one new rule produced. Each generation produces more number of rules so production of number of rules is directly proportional to the number of generations. After some generations, production of rules stops or decreases it is observed.

Fuzzy apriori algorithm improves the accuracy for detecting intrusions. For testing, we used 500 labelled connections having 400 connections of intrusion type and 100 of normal type. After 200 generations, 97% detection rate is observed for misuse intrusion detection.

VI. CONCLUSION

The paper presents fuzzy data mining algorithm using genetic concepts for the Intrusion detection system for detecting 4 major attacks DoS, R2L, U2R, and Probe from KDD99CUP data set. The architecture of the proposed system is discussed. This algorithm provides a high rate of the rule set for detecting different types of attacks. Our system is more flexible for usage in different application areas with proper attack taxonomy. As the intrusions are becoming complex and alter rapidly an IDS should be capable to compete with the thread space.

GA gives good result but not able to handle the sharp boundary problems. So, for dealing with sharp boundary, fuzzy theory is best suited. For increasing accuracy for intrusion detection combination of fuzzy data mining with GA will be more powerful.

REFERENCES

[1] Zhixian Chen, An Approach to Network Misuse Detection Based on Extension Matrix and Genetic Algorithm, Proc. 5th IEEE Int. Conf. on Cognitive Informatics (ICCI'06), 2006, 1-4244-0475-4.

[2] B. Abdullah, I. Abd-alghafar, Performance Evaluation of a Genetic Algorithm Based Approach to Network Intrusion Detection System, 13th International Conference on AEROSPACE SCIENCES & AVIATION TECHNOLOGY, ASAT- 13, 2009.

[3] Ren Hui Gong, M. Zulkernine, P. Abolmaesumi, A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection, Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks, IEEE, 2005.

[4] D. E. Goldberg, Genetic Algorithm in Search Optimization and Machine Learning. Reading, (MA: Addison-Wesley, 1989).

[5] Mei Li, *Using Genetic Algorithm for Network Intrusion Detection*, Department of Computer Science and Engineering Mississippi State University, Mississippi State, MS 39762.

[6] T. Lappas and K. Pelechrinis, *Data Mining Techniques for (Network) Intrusion Detection Systems*, Department of Computer Science and Engineering UC Riverside, Riverside CA 92521.

[7] B. Uppalaiah, K. Anand, Genetic Algorithm Approach to Intrusion Detection System, *International Journal of Computer Sci ence And Technology, Vol. 3, Issue 1*, Jan.-March 2012.

[8] S. Selvakani and R.S.Rajesh, Genetic Algorithm for framing rules for Intrusion Detection, IJCSNS International Journal of Computer Science and Network Security, vol.7 No.11, 285-290, November 2007.

[9] Z. Bankovic, D. Stepanovic, S. Bojanic, Improving Network Security using Genetic Algorithm Approach, *Computer and Electrical Engineering, pp. 438-451, 2007. –b*

[10] Md. Sazzadul Hoque, Md. Abdul Mukit and Md. Abu Naser Bikas, An Implementation of Intrusion Detection System using Genetic Algorithm, *International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.2, March 2012. –d.*

[11] A.A. Ojugo, A.O. Eboka, Genetic Algorithm Rule-Based Intrusion Detection System (GAIDS), Journal of Emerging Trends in Computing and Information Sciences, VOL. 3, NO. 8 Aug, 2012 ISSN 2079-8407.

[12] Ms.Nivedita Naidu, An Effective Approach to Network Intrusion Detection System using Genetic Algorithm, *2010 International Journal of Computer Applications (0975 – 8887) Volume 1 – No. 2.*

[13] Vivek K. Kshirsagar, Intrusion Detection System using Genetic Algorithm and Data Mining: An Overview, *International Journal of Computer Science and Informatics, Vol-1, Iss-4, 2012.*

[14] S. Mabu, Ci Chen, K. Shimada, "An Intrusion-Detection Model Based on Fuzzy Class-Association-Rule Mining Using Genetic Network Programming," IEEE Transactions Systems, Man, Cybernetics C, Application and Reviews, volume 41, number 1, pp. 130–139, January 2011.

[15] J. Luo, "Integrating fuzzy logic with data mining methods for intrusion detection," Master's Thesis, Department of Computer Science, Mississippi State University, Starkville, MS, 1999.

[16] J. G.-P. A. El Semaray, J. Edmonds, and M. Papa, "Applying data mining of fuzzy association rules to network intrusion detection," presented at the IEEE Workshop Information, United States Military Academy, West Point, NY, 2006.

[17] J. Han, M. Kamber, "Data Mining", Morgan Kaufmann Publishers, 2001.

