International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 2, Issue 6, November – December 2013 ISSN 2278-6856

GNP-Based Fuzzy Class-Association Rule Mining in IDS

Manjunath Suryavanshi[1], Bahubali Akiwate[2] and Mallappa Gurav[3]

[1, 2, 3]Dept. Of Computer Science and Engineering,
K. L. E College of Engineering & Technology, Chikodi-591 201, Karnataka, India

Abstract: ***As the Internet services spread all over the world, many kinds of security threats are increasing. Therefore, existing Intrusion Detection Systems (IDS) facing very serious issue for the Internet users for their day to day online transactions, like Internet banking, online shopping, foreign exchange and trading stocks. Genetic Algorithm is used to identify various attacks on different type of connections. This algorithm takes into consideration of different features in network connections such as protocol type, duration, service, to generate a classification rule set. Each rule set identifies a specific type of attacks. A novel fuzzy class-association rule mining method based on Genetic Network Programming (GNP) is used for detecting such network intrusions. By combining fuzzy set theory with GNP, the proposed method can deal with KDDCup99 mixed dataset that contains both discrete and continuous attributes. This method focuses on building distribution of normal and intrusion accesses based on fuzzy GNP. In an application of intrusion detection the training dataset contains both normal connections and several kinds of intrusion connections. GNP examines all the tuples of the connections in the dataset to pick up the rules to be stored in two independent rule pools; normal pool and intrusion pool. Fitness function is defined, higher fitness of a rule results in high Detection Rate (DR) and low Positive False Rate (PFR), which means probability of intrusion is high in the connection. By using this data can be classified into normal class, intrusion class.***

Keywords: Genetic Network Programming, Class-association-rule mining, Fuzzy membership function, Intrusion detection, KDDCup99 dataset.

1. INTRODUCTION

Now a day's many kinds of systems over the Internet such as online shopping, Internet banking, trading stocks and foreign exchange, and online auction have been developed. However, due to the open society of the Internet, the security of our computer systems and data is always at risk. The extensive growth of internet has prompted network intrusion detection to become a critical component of infrastructure protection mechanisms. Network intrusion detection can be defined as identifying a set of malicious actions that threaten the integrity, confidentiality and availability of a network resource. Intrusion detection is traditionally divided into two categories, i.e., misuse detection and anomaly detection. Misuse detection mainly searches for specific patterns or sequences of programs and user behaviors that match well-known intrusion scenarios [1].

In 1987 Dorothy E. Denning proposed intrusion detection as is an approach to counter the computer and networking attacks and misuses [2]. It is highly difficult to provide complete security to the system though we have several protection techniques. In the network accessing and exchanging the information may be easy but providing the security for the information is difficult. Intrusion detection recognizes the unauthorized access to the network, mischievous attacks on the computer systems. To recognize the attacks and detect the intrusions the intrusion detection technology is more useful. Intruders can be classified into two types as External Intruder or Internal Intruder. The unauthorized users who enter the system and make changes to the system and access the resource in the network without authorization, is an external intruder. The intruder in the network without user accounts trying to attack the system is an internal intruder. Intrusion detection systems are classified into two types Misuse detection and Anomaly detection. Intrusion detection with known patterns is called misuse detection. Identifying the abnormalities from the normal network behaviors is called anomaly detection. Hybrid detection systems combine both the misuse and anomaly detection systems. The network traffic and individual packets for mischievous traffic is tested by a network based IDS. An intrusion detection system is a device or software application that monitors network and/or system activities for malicious activities or policy violations and produces reports.

While, anomaly detection develops models of normal network behaviors, and new intrusions are detected by evaluating significant deviations from the normal behavior. KDD99Cup [3] is widely used as training and testing datasets for the evaluation of IDSs. Data mining generally refers to the process of extracting useful rules from large amount of data. The recent rapid development in data mining contributes to developing wide variety of algorithms suitable for network intrusion detection problems. Intrusion detection can be thought of as a classification problem: where each pattern classified as normal or a particular kind of intrusion.

2. INTRUSION DETECTION OVERVIEW

The below sections give a short overview of networking attacks and classifications.

2.1 Networking Attacks

This section is an overview of the four major categories of

networking attacks. Every attack on a network can comfortably be placed into one of these groupings [5].

1) Denial of Service (DoS): A DoS attack is a type of attack in which the hacker makes a computing or memory resources too busy or too full to serve legitimate networking requests and hence denying users access to a machine e.g. apache, smurf, neptune, ping of death, back, mail bomb, UDP storm etc. are all DoS attacks.

2) Remote to User Attacks (R2L): A remote to user attack is an attack in which a user sends packets to a machine over the internet, which he/she does not have access to in order to expose the machines vulnerabilities and exploit privileges which a local user would have on the computer e.g. xlock, guest, xnsnoop, phf, sendmail dictionary etc.

3) User to Root Attacks (U2R): These attacks are exploitations in which the hacker starts off on the system with a normal user account and attempts to abuse vulnerabilities in the system in order to gain super user privileges e.g. perl, xterm.

4) Probing: Probing is an attack in which the hacker scans a machine or a networking device in order to determine weaknesses or vulnerabilities that may later be exploited so as to compromise the system. This technique is commonly used in data mining e.g. saint, portsweep, mscan, nmap etc.

2.2 Classification of Intrusion Detection

Intrusions Detection can be classified into two main categories. They are as follow:

1) Host Based Intrusion Detection: HIDSs evaluate information found on a single or multiple host systems, including contents of operating systems, system and application files.

2) Network Based Intrusion Detection: NIDSs evaluate information captured from network communications, analyzing the stream of packets which travel across the network.

As host-based systems rely heavily on audit trails, they become limited by these audit trails, which are not provided by the manufacturers who design the intrusion detection system itself. As a result, theses trails may not necessarily support the needs of the intrusion detection system.

Network-based intrusion detection systems offer a different approach. These systems collect information from the network itself, rather than from each separate host. They operate essentially based on a wiretapping concept; information is collected from the network traffic stream, as data travels on the network segment. The intrusion detection system checks for attacks or irregular behavior by inspecting the contents and header information of all the packets moving across the network [6].

3. GNP-BASED FUZZY CLASS ASSOCIATION RULE MINING

A class-association rule mining algorithm based on GNP has been proposed [7]. GNP and its class association rule mining are briefly explained as follows.

3.1 Framework of Genetic Network Programming

GNP is one of the evolutionary optimization techniques, which uses directed graph structures instead of strings and trees. The phenotype and genotype expressions of GNP are shown in Figure 1.

GNP is composed of three types of nodes: start node, judgment node, and processing node. Judgment nodes, J_1, J_2,…, J_m (m is the total number of judgment functions), serve as decision functions that return judgment results so as to determine the next node. Processing nodes, P_1, P_2…, P_n (n is the total number of processing functions), serve as action/processing functions. The practical roles of these nodes are predefined and stored in the function library by supervisors. Once GNP is booted up, the execution starts from the start node, then the next node to be executed is determined according to the connection between nodes and a judgment result of the current activated node.

Figure 1 also describes the gene of a node in a GNP individual. NT_i represents the node type such as 0 for start node, 1 for judgement node and 2 for processing node. ID_i serves as an identification number of a judgement or processing node, for example, $NT_i=1$ and $ID_i=2$ represents node function J_2. C_{i1}, C_{i2}…., C_{ij} denote the node numbers connected from node i. The total number of nodes in an individual remains the same during every generation [1].

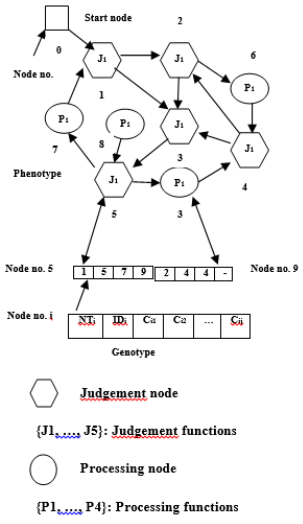

Figure 1 Basic Structure of GNP individuals

Three kinds of genetic operators, i.e., selection, mutation, and crossover, are implemented in GNP.

1) Selection: Individuals are selected according to their fitness.

2) Crossover: Two new offspring are generated from two parents by exchanging the genetic information. The selected nodes and their connections are swapped each other by crossover rate P_C.

3) Mutation: One new individual is generated from one original individual by the following operations. Each node branch is selected with the probability $Pm1$ and reconnected to another node. Each node function is selected with the probability $Pm2$ and changed to another one.

3.2 GNP-Based Class-Association Rule

A judgment node in GNP has a role in checking an attribute value in a tuple [6]. Candidate class-association rules are represented by the connection of judgement nodes. An example of the representation is shown in Figure 2.

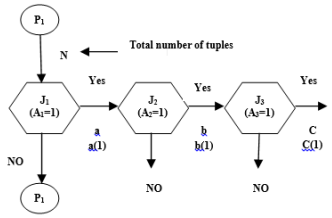

Figure 2 Node transition to find class-association rules

Processing node P_1 serves as the beginning of class-association rules. A_1=1, A_2=1, and A_3=1 denote the judgment functions. If a tuple satisfies the condition of the judgment function, Yes-side branch is selected and the condition of the next judgment function is examined in order to find longer rules. No-side is connected to processing node P_2 to start examining other rules. Therefore, the branch from the judgment node represents the antecedent part of class-association rules, while the fixed consequent part can be predefined.

For example, the class-association rules such as

$$(A_1 = 1) \Rightarrow (C = 1)$$
$$(A_1 = 1) \wedge (A_2 = 1) \Rightarrow (C = 1)$$
$$(A_1 = 1) \wedge (A_2 = 1) \wedge (A_3 = 1) \Rightarrow (C = 1)$$
$$(A_1 = 1) \Rightarrow (C = 0)$$
$$(A_1 = 1) \wedge (A_2 = 1) \Rightarrow (C = 0)$$
$$(A_1 = 1) \wedge (A_2 = 1) \wedge (A_3 = 1) \Rightarrow (C = 0)$$

are examined by the node transition in Figure 2.

The procedure of examining tuples is as follows. The first tuple in the database is read and the node transition starts from processing node P_1. Then, if Yes-side branch is selected, the current node is transferred to the next judgment node. If No-side branch is selected, the current node is transferred to processing node P_2 to find other rules. The same procedure is repeated until the node transition started from the last processing node P_n is finished. After examining the first tuple in the database, the second tuple is read and the node transition starts from processing node P_1 again. Finally, all the tuples are examined by repeating the above node transitions. Note that the number of judgment functions (J_1, J_2,…) equals the number of attributes (A_1, A_2,…) in the database.

3.3 Sub attributes Utilization

Network connection data have their own characteristics, such as discrete and continuous attributes, and these attribute values are important information that cannot be lost. We introduce a sub attribute-utilization mechanism concerning binary, symbolic and continuous attributes to keep the completeness of data information. Binary attributes are divided into two sub attributes corresponding to judgment functions. For example, binary attribute $A1$ (=land) was divided into A_11 (representing land=1) and A_12 (representing land=0). The symbolic attribute was divided into several sub attributes, while the continuous attribute was also divided into three sub attributes concerning the values represented by linguistic terms (low, middle, and high) of fuzzy membership functions predefined for each continuous attribute. Figure 3 shows a division example of the three attributes.

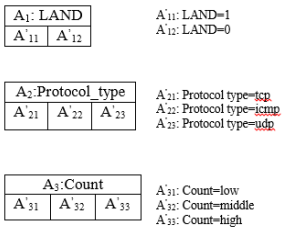

Figure 3 Example of sub attributes utilization

In the conventional GNP-based class-association rule mining, only discrete attributes with value 1 are considered. In the proposed method, all the values such as 0 and 1 for binary attributes and text values for symbolic attributes are considered.

3.4 Rule Extraction by GNP with Fuzzy Membership Functions

GNP examines the attributes of tuples at judgment nodes and calculates the measurements of association rules at processing nodes [7]. Judgment nodes judge the values of the assigned sub attributes, e.g., Land=1, Protocol=tcp, etc. The GNP-based fuzzy class-association rule mining with sub attribute utilization successfully combines discrete and continuous values in a single rule. An example of the node transition in the proposed method is shown in Figure 4.

P_1 is a processing node that serves as a starting point of class association rules and connects to a judgment node. The Yes-side of the judgment node is connected to another judgment node, while the No-side is connected to the next processing node. Judgment nodes shown here have the functions that examine the sub attributes including both discrete and continuous attributes.

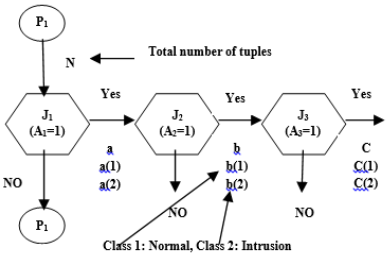

Figure 4 Example of node transition in fuzzy class-association rule mining based on GNP

In Figure 4, judgment node J_1 examines the value of the binary sub attribute land=1, J_2 examines the value of the symbol sub attribute protocol=tcp, and J_3 examines the fuzzy membership value of the continuous sub attribute count=Low. In the case of binary and discrete attributes (J_1 and J_2), GNP selects Yes-side branch and goes to the next judgment node if "land=1 is Yes" or "count=Low is Yes," otherwise, the current node is transferred to processing node P_2 to start examining other rules. In the case of continuous attribute ($J3$), after calculating the fuzzy membership value of the sub attribute, the value is regarded as a probability of selecting Yes-side branch. When No-side branch is selected, the current node is transferred to processing node P_2. The total number of tuples moving to Yes-side at each judgment node is memorized in the processing node from which rule extraction starts. In Fig. 4, N is the number of total tuples in the database, and a, b, and c are the number of tuples moving to Yes-side at each judgment node. In an application of misuse detection, the training database contains both normal connections and several kinds of intrusion connections. Thus, GNP examines all the tuples of the connections in the database and counts the numbers a, b, c, $a(1)$, $b(1)$, $c(1)$, $a(2)$, $b(2)$, and $c(2)$, where a, b, and c are the numbers of tuples moving to Yes-side at the judgment nodes, $a(1)$, $b(1)$, and $c(1)$ are those with class C=1 (normal) and $a(2)$, $b(2)$, and $c(2)$ are those with class C =2 (intrusion).

4. RESULTS

GNP with fuzzy data mining is compared with crisp data mining, and the result clarifies the necessity to introduce fuzzy membership functions into GNP-based data mining. GNP can extract many rules of normal connections and known intrusion connections from the training database. When we use them for intrusion detection, the matching of a new connection with the normal rules and the intrusion rules are calculated, respectively.

Table 1
Result of Crisp Data Mining with K=0.5 Intrusion Detection

	Normal (T)	Intrusion (T)	Total
Normal (C)	150	44	194
Intrusion (C)	4	575	579
Total	154	619	773

Table 2
Result of proposed method with K=0.5 Intrusion Detection

	Normal (T)	Intrusion (T)	Total
Normal (C)	181	13	194
Intrusion (C)	15	564	579
Total	196	577	773

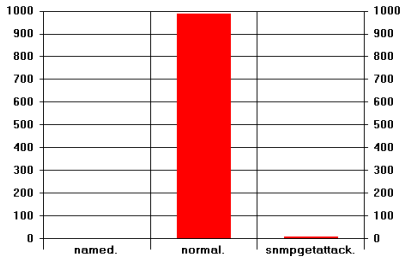

Figure 5 Attack Detection

From the above experiment, it is able to create a rule that could successfully classify all 773 sample network connections. Along with this, it also classifies 181 normal connections and 13 as network connection attacks among 194 normal connections. For the same intrusion detection connections 579 our Intrusion Detection System(IDS) classifies 15 as normal connection and 564 as intrusion connection. Figure 6 shows different parameter analysis for sample data.

Cutoff Value	True Positives	True Negatives	False Positives	False Negatives
0.01000	41	0	59	0
0.02000	41	17	42	0
0.03000	41	31	28	0
0.04000	41	42	17	0
0.68000	41	53	6	0
0.68000	41	59	0	0
0.67000	41	59	0	0
0.65000	41	59	0	0
0.66000	41	59	0	0
0.70000	41	59	0	0

Figure 6 Different parameters Analysis

5. CONCLUSION

The paper represents an efficient Intrusion-Detection model based on Fuzzy Class-Association rule mining

using Genetic Network Programming from KDD99CUP data set. Discrete and continuous attributes are consistently used to extract many good rules for classification. In future Deterministic Finite State Automata (DFA) can be used on binary attributes to detect intrusion of network connections.

References

[1] Shingo Mabu, Ci Chen, Nannan Lu, Kaoru Shimada, and Kotaro Hirasawa, "An Intrusion-Detection Model Based on Fuzzy Class-Association-Rule Mining Using Genetic Network Programming", IEEE Transactions On Systems, Man, And Cybernetics-Part C: Applications And Reviews, VOL. 41, NO. 1, January 2011

[2] B. Uppalaiah, K. Anand, B. Narsimha, S. Swaraj, T. Bharat, "Genetic Algorithm Approach to Intrusion Detection System", IJCST Vol. 3, Issue 1, Jan.-March 2012

[3] Kddcup 1999 data [Online]. Available: kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[4] M. Crosbie and G. Spafford, "Applying genetic programming to intrusion detection", presented at the AAAI Fall Symp. Series, AAAI Press, Menlo Park, CA, Tech. Rep. FS-95-01, 1995.

[5] Mohammad Sazzadul Hoque1, Md. Abdul Mukit2 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.

[6] Bace, Rebecca Gurley: *Intrusion Detection*. Copyright 2000 by Macmillan Technical Publishing, ISBN 1-57870-185-6.

[7] K. Shimada, K. Hirasawa, and J. Hu, "Genetic network programming with acquisition mechanisms of association rules", J. Adv. Comput. Intell. Intell. Inf., vol. 10, no. 1, pp. 102–111, 2006.

[8] 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 Inf., United States Military Academy, West Point, NY, 2006.

AUTHORS

Mr. Manjunath Suryavanshi received a Bachelor of Engineering degree in Computer Science and Engineering from Gogte Institute of Technology, affiliated to VTU, Belgaum, during the year 2009. He completed M.Tech degree in Software Engineering from M. S. Ramaiah Institute of Technology, an autonomous institute affiliated to VTU, Belgaum, during the year 2013. He is working as an assistant professor in Computer Science and Engineering Department of K. L. E. College of Engineering and Technology, Chikodi, since from August-2011.

Mr. Bahubali Akiwate received a Bachelor of Engineering degree in Computer Science and Engineering from Bahubali College of Engineering, Shravanabelagola, affiliated to VTU, Belgaum, during the year 2009. He completed M.Tech Degree in Digital communication and Networking from Gogte Institute of Technology, Belgaum, from the same University, during the year 2011. He is working as an assistant professor in Computer Science and Engineering Department of K. L. E. College of Engineering and Technology, Chikodi since from September-2011.

Mr. Mallappa Gurav received the Bachelor of Engineering degree in Computer Science and Engineering from Basaveshwar Engineering College, Bagalkot, affiliated to VTU, Belgaum, during 2010. He is persuing M.Tech at Gogte Institute of Technology, Belgaum, under Visvesvaraya Technological University, Belgaum. Currently he is working as an assistant professor in Computer Science and Engineering Department of K. L. E. College of Engineering and Technology, Chikodi since from 2010.