Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Network Intrusion Detection using Fuzzy Class Association Rule Mining Based on Genetic Network Programming

Ci Chen*, Shingo Mabu*, Chuan Yue*, Kaoru Shimada*,and Kotaro Hirasawa*
*Graduate School of Information, Production and Systems, Waseda University,
Hibikino 2-7, Wakamatsu-ku, Kitakyusyu, Fukuoka 808-0135, Japan
Email: chenci@ruri.waseda.jp, mabu@aoni.waseda.jp, libra1009@toki.waseda.jp, k.shimada@aoni.waseda.jp, hirasawa@waseda.jp

***Abstract*—Computer Systems are exposed to an increasing number and type of security threats due to the expanding of internet in recent years. How to detect network intrusions effectively becomes an important techniques. This paper presents a novel fuzzy class association rule mining method based on Genetic Network Programming(GNP) for detecting network intrusions. GNP is an evolutionary optimization techniques, which uses directed graph structures as genes instead of strings(Genetic Algorithm) or trees(Genetic Programming), leading to creating compact programs and implicitly memorizing past action sequences. By combining fuzzy set theory with GNP, the proposed method can deal with the mixed database which contains both discrete and continuous attributes. And it can be flexibly applied to both misuse and anomaly detection in Network Intrusion Detection Problem. Experimental results with KDD99Cup and DAPRA98 databases from MIT Lincoln Laboratory show that the proposed method provides a competitively high detection rate compared with other machine learning techniques.**

***Index Terms*—network intrusion detection, fuzzy membership function, class association rule mining, Genetic Network Programming**

I. Introduction

The security of our computer systems and data is always at risk. The extensive growth of the 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. Traditionally, intrusion detection is divided into 2 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. While, anomaly detection develops models of normal network behaviors, and new intrusions are detected by evaluating significant deviations from the normal behavior. The advantage of anomaly detection is that it may detect novel intrusions that have not been observed yet.

In this paper, we propose a fuzzy class association rule mining approach based on Genetic Network Programming(GNP) to apply to both misuse detection and anomaly detection. For misuse detection, the normal pattern rules and intrusion pattern rules are extracted from the training dataset. Classifiers are build up according to these extracted rules. While for anomaly detection, we focus on extracting as many normal pattern rules as we can. Extracted rules are used to detect novel or unknown intrusions by evaluating the deviation from the normal behavior.

We have already proposed the class association rule mining algorithm based on GNP[1]. In this paper, we extend and improve the conventional method by combining the fuzzy set theory so that it can deal with both the discrete and continuous attributes in one database which is a normal situation in real world applications. In addition to that, a sub-attribute utilization method is proposed so as to avoid losing data during the rule extraction. And also specific classification methods are designed respectively for two kinds of detection. The features of the proposed method are as follows:

- Proposed fuzzy class association rule mining method can deal with both discrete and continuous attributes in the database, which is practically useful for real network-related databases;
- Proposed sub-attribute utilization method considers all attribute values as information, which contributes to avoiding data loss;
- Proposed fitness function provides the flexibility of mining more new rules or mining rules with higher accuracy;
- Proposed method can be flexibly applied to both misuse detection and anomaly detection with specific-designed classifiers.

The paper is organized as follows: The background information of Genetic Network Programming and class association rule mining is overviewed in section 2, followed by the introduction of the proposed method applied to misuse and anomaly detection in detail in section 3. Section 4 describes the specific classification method by GNP-based class association rules. And simulation results with two databases KDD99Cup and DAPRA98 are given in section 6. Finally, conclusions and future work are mentioned in section 7.

II. Background

We have already proposed the class association rule mining algorithm based on GNP in the previous research[1]. In this

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

section, the outline of Genetic Network Programming(GNP) and class association rule mining based on GNP is briefly reviewed.

A. Genetic Network Programming[2][3][4][5]

GNP is one of the evolutionary optimization techniques, which uses directed graph structures as genes instead of strings and trees. The basic structure of GNP is shown in Fig.1. GNP is composed of three types of nodes: start node, judgment node and processing node. The judgment nodes, the set of $J_1, J_2, ..., J_m$, serve as decision functions, which return judgment results so as to determine the next node. While, processing nodes, the set of $P_1, P_2, ..., P_n$, serve as action/processing functions.

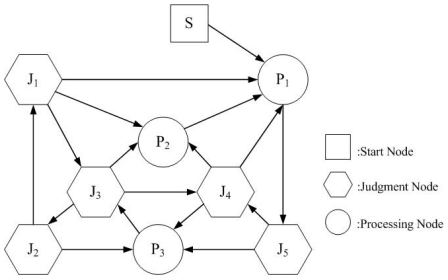

Fig. 1. **Basic Structure of GNP Individual**

Three kinds of genetic operators, selection, mutation and crossover, are used in GNP.

B. Class Association Rule Mining[6]

The following is a statement of mining association rules[6]. Let $I = \{A_1, A_2, ..., A_l\}$ be a set of literals, called items or attributes. Let G be a set of transactions, where each transaction *T* is a set of items such that $T \subseteq I$. Associated with each transaction is a unique identifier whose set is called TID. A transaction *T* contains *X*, a set of some items in *I*, if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $U \subset I$, and $X \cap Y = \emptyset$. *X* is called antecedent and *Y* is called consequent of the rule. If the fraction of transactions containing *X* in *G* equals x, then we say that $support(X) = x$. The Rule $X \Rightarrow Y$ has a measure of its strength called confidence defined as the ratio of $support(X \cup Y)/support(X)$.

Let A_i be an attribute(item) in a database with value 1 or 0, and *k* be the class labels. Class Association Rule can be represented by

$(A_p = 1) \wedge ... \wedge (A_q = 1) \Rightarrow (C = k) \quad k \in \{0, 1\}$, as a special case of the association rule $X \Rightarrow Y$ with fixed consequent *C*.

Calculation of χ^2 value of rule $X \Rightarrow Y$ is shown as follows. Assume $support(X) = x, support(Y) = y, support(X \cup Y) = z$ and the total number of tuples is N. We can calculate χ^2 as

$$\chi^2 = \frac{N(z - xy)^2}{xy(1-x)(1-y)}. \quad (1)$$

If χ^2 is higher than a cutoff value, we should reject the assumption that *X* and *Y* are independent (3.84 at the 95% significance level or 6.64 at the 99% significance level).

C. GNP-based Class Association Rule[1]

Attributes and its values are corresponding to the functions of the judgment nodes in GNP. The connections of judgment nodes are represented as class association rules. An example of the representation is shown in Fig.2. P_1 is a processing node, which serves as the beginning of class association rules. $A_1 = 1, A_2 = 1$ and $A_3 = 1$ denote the functions of judgment nodes. The connection represents the antecedent part of class association rules, for the fixed consequent part can be defined in advance.

For example, the class association rules like
$(A_1 = 1) \Rightarrow (C = 1)$,
$(A_1 = 1) \wedge (A_2 = 1) \Rightarrow (C = 1)$,
$(A_1 = 1) \wedge (A_2 = 1) \wedge (A_3 = 1) \Rightarrow (C = 1)$,
$(A_1 = 1) \Rightarrow (C = 0)$,
$(A_1 = 1) \wedge (A_2 = 1) \Rightarrow (C = 0)$,
$(A_1 = 1) \wedge (A_2 = 1) \wedge (A_3 = 1) \Rightarrow (C = 0)$ are represented by the connection in Fig.2. The measurements include *support*, *confidence* and χ^2 *value*. Judgment node determines the next node by the judgment result. If the attribute which the judgment represents is satisfied, then it moves to another judgment node (Yes side). Otherwise, it moves to another processing node (No side). In Fig.2, *N* is the number of total tuples. a, b, c, a(1), b(1) and c(1) are the numbers of tuples moving to the Yes-side at each judgment node and the number of tuples moving to the yes-side at each judgment node on the condition of Class 1, respectively.

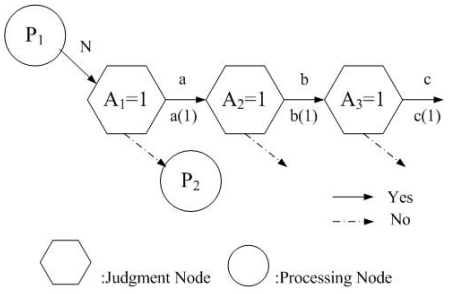

Fig. 2. **GNP Representation of Class Association Rules**

III. GNP-based Fuzzy Class Association Rule Mining with Sub-attribute Utilization

In this section, the proposed methods for both misuse detection and anomaly detection using improved GNP-based fuzzy class association rule mining are described in details.

A. Data Preprocessing

The DARPA98 training data includes "list file" which identifies each network connections' time stamps, service type, source IP address, source port, destination IP address, destination port and the type of each attack. However, some intrinsic

features and time-based features of the network connection [7] which are important to intrusion detection haven't been included.

We used tcptrace utility software [8] to extract information about packets to construct new intrinsic features such as data bytes, SYN and FIN packets flowing from source to destination as well as from destination to source. This information was combined with the original list file by matching the time stamp to form the new network connection data. This technique was also used in constructing KDDCup99 data [9], but this dataset lacks time information. After getting more intrinsic features for each connection, we used a time window of 2 seconds to get time-based features that count various characteristics in previous 2 second connections.

After the data preprocessing for each network connection, there are 30 attributes including both the list file features, intrinsic features and time based features.

Another database KDD99CUP originally includes 41 attributes, which doesn't need Data Preprocessing.

B. Sub-Attribute Utilization

Network connection data have its own characteristics, such as containing both discrete and continuous attributes, and all attribute values are important information that cannot be lost.

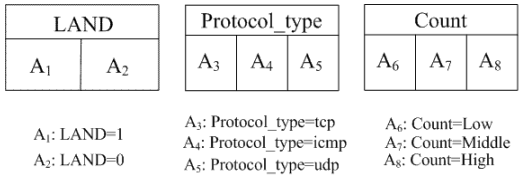

Fig. 3. **An Example of Sub-Attribute Utilization**

We propose a sub-attribute utilization mechanism concerning binary, symbolic and continuous attributes to keep the completeness of data information. Binary attributes was divided into 2 sub-attributes corresponding to judgment functions. For example, the binary attribute land was divided into A_1(representing land=1) and A_2(representing land=0). The symbolic attribute was divided into several sub-attributes, while the continuous attribute was also divided into 3 sub-attributes concerning the value represented by linguistic terms (Low, Middle, and High) generated by fuzzy membership functions predefined for each continuous attributes. Details about this method will be discussed in the next section. Fig.3 shows 3 different examples of the attribute division.

C. Fuzzy Membership Function for Continuous Attributes

All values of continuous attributes in the database are transformed into 3 linguistic terms (Low, Middle and High). Each continuous attribute is combined with its own membership function predefined. These linguistic terms are obtained using the fuzzy membership functions spaced symmetrically and equally as shown in Fig.4.

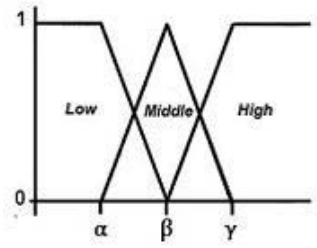

Fig. 4. **Definition of the fuzzy membership function**

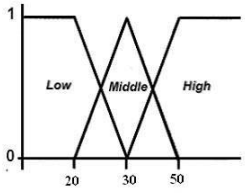

Fig. 5. Membership function for A_1

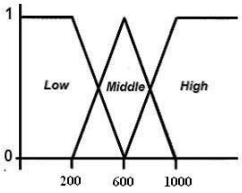

Fig. 6. Membership function for A_2

The parameters α, β and γ for fuzzy membership functions are shown in Fig.4. The parameters of the membership function for Attribute *A* are set as follows:

β =average value of attribute *A* in the database;

2γ =largest value of attribute *A* in the database;

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

Table.1 shows an example of a small database with two continuous attributes. And Fig.5. and Fig.6. shows the corresponding predefined fuzzy membership functions respectively for each attribute. Finally, Table.2 shows the database with fuzzy membership values after transformation using the predefined membership functions.

When one judgment node in GNP represents an continuous

TABLE I
AN EXAMPLE OF A SMALL DATABASE

TID	A_1	A_2
1	10	1000
2	20	800
3	30	600
4	20	400
5	10	200

TABLE II
DATABASE WITH FUZZY MEMBERSHIP VALUES

TID	Attribute A_1			Attribute A_2		
	Low	*Mid*	*High*	*Low*	*Mid*	*High*
1	1.0	0	0	0	0	1.0
2	1.0	0	0	0	0.5	0.5
3	0	1.0	0	0	1.0	0
4	0	0.5	0.5	0.5	0.5	0
5	0	0	1.0	1.0	0	0

attribute *A* with linguistic terms q_i, the fuzzy membership value is employed to determine the transition from the judgment node to the next node. The detailed method is generating a random number range from 0 to 1, and comparing it to the membership value of the fuzzy attribute *A* with that linguistic terms q_i in the database, which serves as the probability of going to the Yes-side of the transition.

D. GNP Structure for Combined Association Rule Mining

GNP examines the attribute values of database tuples using judgment nodes and calculates the measurements of association rules using processing nodes[1]. Attributes and their values correspond to judgment nodes and their judgment value in GNP, respectively. The conventional representation of the class association rule using GNP is already shown in Fig.2. The fuzzy class association rule mining based on GNP with sub-attribute utilization successfully combines discrete and continuous values in one single rule. The example of the new representation is shown in Fig.7.

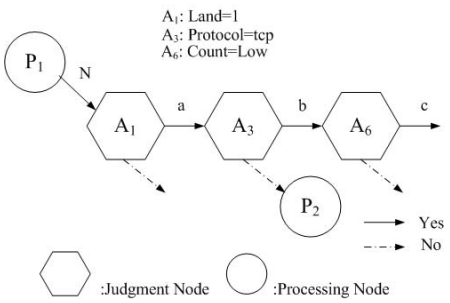

Fig. 7. **GNP Representation of Combined Class Association Rule Mining with Sub-Attribute Utilization**

P_1 is a processing node, which serves as a starting point of class association rules and connects to a judgment node. The Yes-side of a judgment node is connected to another judgment node, while the No-side is connected to the next processing node. Judgment nodes here are corresponding to the sub-attributes including both discrete and continuous attributes. Taking the above as an example, judgment node A_1 represents the value of the binary attribute *land* equals to *one*; A_3 represents the value of the symbol attribute *protocol* belongs to *tcp* and A_6 represents the fuzzy membership value of the continuous attribute *count* equals to *Low*. For binary or discrete attributes, GNP goes to the next judgment node if that A_1 equals to *one* or A_3 equals to *tcp* is satisfied or goes to the processing node if it is denied. However, for continuous attributes, a random number is generated and compared to the fuzzy membership value with linguistic term *Low* of one tuple in the database. If it is smaller that that membership value, GNP goes to the next judgment node, otherwise, it goes to another processing node to start a new rule.

E. Rule Extraction

The total number of tuples moving to Yes-side at each judgment node is calculated at each processing node, which is a start point for association rule mining. In Fig.7, *N* is the number of total tuples, and *a, b, c* are the number of tuples moving to Yes-side at each judgment node.

For the specific application of misuse detection, the training database contains both normal connections and several kinds of intrusion connections. Thus, we check all the tuples of the connections in the database and count the number *N, a, b, c,* a_n, b_n, c_n, a_i, b_i and c_i, for both classes, normal class and intrusion class, respectively. Then, the criteria of $sup > 0.25$, $conf > 0.6$ and $\chi^2 > 6.64$ are used to pick up the rules to be stored in two independent rule pools, normal rule pool and intrusion rule pool. Table.3 shows the calculation of support and confidence values of class association rules involved in Fig.7. χ^2 can be calculated according to Eq.(1).

TABLE III
MEASURES OF CLASS ASSOCIATION RULES

Rules	Support	confidence
$(Land = 1) \Rightarrow Normal$	a_n/N	a_n/a
$(Land = 1) \Rightarrow Intrusion$	a_i/N	a_i/a
$(Land = 1) \wedge (Pro = tcp) \Rightarrow Normal$	b_n/N	b_n/b
$(Land = 1) \wedge (Pro = tcp) \Rightarrow Intrusion$	b_i/N	b_i/b
$(Land = 1) \wedge (Pro = tcp) \wedge (Count = Low) \Rightarrow Normal$	c_n/N	c_n/c
$(Land = 1) \wedge (Pro = tcp) \wedge (Count = Low) \Rightarrow Intrusion$	c_i/N	c_i/c

For another application of anomaly detection, the training database we used is normal connection, in this case, only support is used as the criteria to extract important normal class association rules.

Rules with support values greater than 0.5 are extracted as important class association rules for normal behaviors in the belief that frequent normal patterns should occur more than one half. These rules are stored in the normal rule pool.

F. Fitness and Genetic Operator

The following part is to describe the fitness function and genetic operators used to evolve the GNP individuals in network intrusion detection problem.

Before defining the fitness value of an individual, let's first define the fitness value of an obtained rule. The fitness value of rule *r* is as follows:

$$fitness_r = \frac{Nt_c}{Nt} - \frac{Nn_i}{Nn}, \tag{2}$$

where,

Nt_c: Number of connections correctly detected by rule *r*;

Nt: Number of connections in the training database;

Nn_i: Number of normal connections incorrectly detected by rule *r*;

Nn: Number of normal connections in the training database;

Each rule obtained is checked by training database to get the fitness value. The scale of the fitness value is [-1,1]. A higher fitness value of rules results in high Detection Rate(DR) and low Positive False Rate(PFR) which means the rate of incorrectly assigning normal connections to intrusion. On the other hand, a lower fitness value results in low DR and high PFR.

When an important class association rule is extracted by GNP, the overlap of the attributes is checked to determine whether a rule is new or not. The fitness of GNP individual is defined for network intrusion problems by

$$F = \sum_{r \in R} \{w_1 * fitness_r + w_2 * \alpha_{new}(r)\}, \quad (3)$$

where,
R: set of suffixes of extracted important association rules in a GNP individual;
$finess_r$: the fitness value of rule *r*
$\alpha_{new}(r) = \begin{cases} \alpha_{new} & if\ \ rule\ \ r\ \ is\ \ new \\ 0 & otherwise \end{cases}$

w_1 and w_2 here are the control parameters. On the case of misuse detection, for the training database contains both the normal and intrusion connections, we emphasis more on mining high accuracy rules rather than a large amount of rules, thus a relatively high w_1 value and low w_2 value is reasonable. However, for anomaly detection, the training database contains only the normal connection, and we are eager to find as much normal rules as possible so as to explore the normal space, which leads to a relatively low w_1 value and high w_2 value. The effect of these control parameters will be discussed in detail in the simulation part.

In each generation, individuals are replaced with the new ones by using the following genetic operators so as to obtain more class association rules:

- *Crossover*: In this paper, we use uniform crossover. Judgment nodes are selected as crossover nodes with crossover rate. Two parents exchange the genes of corresponding nodes.
- *Mutation-1*: The connection of judgment nodes is changed with mutation rate1.
- *Mutation-2*: The function of judgment nodes is changed with mutation rate2.

Individuals are ranked by their fitness values after one generation, and top 1/3 individuals with higher fitness values are selected and the offspring is reproduced three times by the above genetic operators for the next generation.

IV. Classification Method by Fuzzy GNP-based Class Association Rules

As this Fuzzy GNP-based Class Association approach is designed for databases containing both discrete and continuous attributes as Network Connection Database, specific classification method is described as follows:

The definition of the matching degree between the continuous attribute A_i in rule *r* with linguistic term q_i and testing data connection with value a_i is :

$$MatchDegree(q_i, a_i) = F_{q_i}(a_i), \quad (4)$$

where, F_{q_i} represents the membership function for linguistic term q_i.

And the matching between rule *r* (p continuous attributes and q discrete attributes) and new unlabeled connection *d* is defined as:

$$Match_r(d) = \frac{1}{p+q}(\sum_{i \in A_p} MatchDegree(q_i, a_i) + t), \quad (5)$$

where,
i: index of continuous attributes in rule *r*;
A_p: set of suffixes of continuous attributes in rule *r*;
p: number of continuous attributes in rule *r*;
q: number of discrete attributes in rule *r*;
t: number of discrete attributes in new unlabeled connection *d* satisfying rule *r*.

$Match_r(d)$ ranges from 0 to 1. If $Match_r(d)$ equals to 1.0, rule *r* matches connection data *d* completely, while $Match_r(d)$ equals to 0, rule *r* does not match connection data *d* at all. Then, the average matching between connection data *d* and all the rules in a certain rule pool is defined as:

$$MATCH(d) = \frac{1}{|R_p|} \sum_{r \in R_p} Match_r(d), \quad (6)$$

where, R_p is the set of suffixes of extracted important class associaiton rules in a certain rule pool.

A. Classifier for misuse detection

The average matching between connection data *d* and all the rules in the normal rule pool $MATCH_n(d)$ and the average matching between connection data *d* and all the rules in the intrusion rule pool $MATCH_i(d)$ are calculated and compared. If $MATCH_n(d) \geq MATCH_i(d)$, connection data *d* is labeled as normal. On the other hand, if $MATCH_n(d) < MATCH_i(d)$, connection data *d* is labeled as intrusion.

In summary, a new connection data is labeled according to their matching with normal and intrusion rule pools. Larger matching suggests the higher possibility of belonging to this class.

B. Classifier for anomaly detection

After getting the matching between each connection data and the rules in the normal rule pool, we can have the distribution of the matching with mean value μ and standard deviation σ. Fig.8 shows one example of the distribution.

In the testing period, when a new unlabeled connection data comes, the matching between the data and the rules in the normal rule pool is calculated. If $MATCH_n(d) < (\mu - k\sigma)$, label the connection as intrusion. On the hand, if $MATCH_n(d) \geq (\mu - k\sigma)$, label it as normal. By adjusting parameter *k*, we can balance the PFR(positive false rate) and NFR(negative false rate).

In all, by using the improved fuzzy GNP-based class association rule mining, we can find a large number of rules

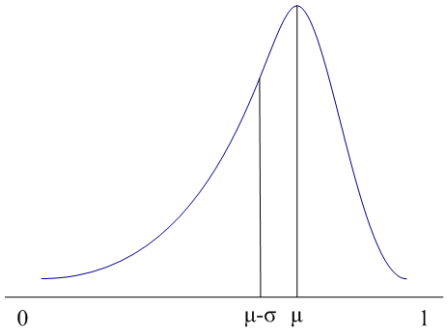

Fig. 8. **An Example of the Distribution of the Matching**

related to normal behaviors so as to explore the space of the normal connections, and any significant deviation from the normal space is viewed as an intrusion.

V. Simulation Results

Parameters setting of GNP is the same for two simulations, i.e., misuse detection and anomaly detection and shown in Table.4.

TABLE IV
Parameters Setting of GNP

Population Size	120
Generation	1000
Processing Node	10
Judgment Node	100
Crossover Rate	1/5
Mutation Rate1	1/3
Mutation Rate2	1/3

A. Simulation.1 for misuse detection

The proposed method for misuse detection is carried out-with KDD99Cup database so as to compared with other machine learning methods.

The training dataset contains 3342 connections randomly selected from KDD99Cup database, among which 1705 connections are normal and other 1637 connections are intrusion, where 3 types of attacks(nepture, smurf and portsweep) are included. 41 attributes are included in each connection, however, after attribute division step, 113 sub-attributes are corresponding to the judgement functions in GNP. 3353 rules in all are extracted after 1000 generations. Fig.9 shows the number of rules extracted versus generation number in simulation.l.

The testing database contained 750 unlabeled normal connections and 240 unlabeled intrusion connections (the same types as training database). Detection results are shown in the Table.5 through the misuse detection classifier we proposed, where, *T* represents testing label and *R* represents real label.

Three criteria[9] are used to evaluate our testing results, DR*(detection rate)*, PFR*(positive false rate)*, and NFR*(negative false rate)*. DR means the total detection rate; PFR means the rate we labeled normal data as intrusion; and NFR means we labeled intrusion data as normal.

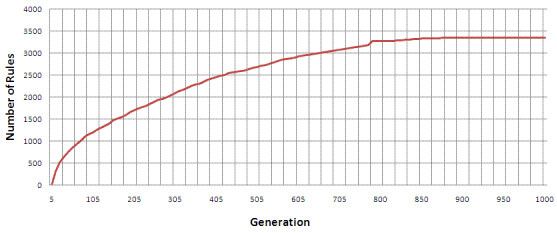

Fig. 9. **Number of Extracted Rules in simulation.1**

TABLE V
Testing Results of the Proposed Method for Misuse Detection

	Normal(T)	Intrusion(T)	Total
Normal(R)	746	4	750
Intrusion(R)	9	231	240
Total	755	235	990

$$DR = (746 + 231)/990 = 98.7\% \quad (7)$$

$$PFR = 4/750 = 0.53\% \quad (8)$$

$$NFR = 9/240 = 3.75\% \quad (9)$$

Compared with other machine learning techniques dealing with KDD99Cup shown in Table.6, we can find out that our proposed method for misuse detection provides higher DR(detection rate) than most of the machine learning techniques except the combination method of SVM with GA or Fuzzy Logic. And in the aspect of PFR(positive false rate), our method also shows a competitive result. In the future work, we may combine our method with SVM to see whether it can improve the performances.

B. Simulation.2 for anomaly detection

The proposed method for anomaly detection is evaluated by carrying out the simulations with DARPA98 database from MIT Lincoln Laboratory[10]. The training database is intrusion-free for the purpose of anomaly detection. It consists of 9137 normal network connection records provided by DARPA98. After preprocessing, 30 attributes are included in every connection record. However, after attribute division, 82 sub-attributes are corresponding to judgment functions in GNP. In 1000 generations, 5589 rules related to normal connections are extracted. Fig.10 shows the number of rules extracted versus generation number, which indicates that the proposed method can extract rules related to normal connections very efficiently.

The testing database consists of 773 connection records including 194 unlabeled normal records and 579 unlabeled intrusion records provided by DAPRA98. Because the training database is intrusion-free, so all 9 kinds of intrusions such as back, ipsweep, land, neptune, pod, port sweep, satan, smurf

TABLE VI
MEASURES OF CLASS ASSOCIATION RULES USING MACHINE LEARNING TECHNIQUES

Technique	DR (%)	PFR(%)
C4.5	95.00	1.00
SVM	95.50	1.00
MLP	94.50	1.00
K-NN	92.00	1.00
LPM	94.00	1.00
RDA	94.00	1.00
FD	89.00	1.00
$\gamma - algorithm$	80.00	1.00
$\kappa - meansclustering$	65.00	1.00
Single leakage clustering	69.00	1.00
Quarter-sphere SVM	65.00	1.00
Y-means clustering	89.89	1.00
Genetic Programming	91.00	0.43
SVM+GA	99.00	-
SVM+Fuzzy Logic	99.56	0.44
Neural Networks+PCA	92.22	-
C4.5+PCA	92.16	-
GA	97.47	0.69
C4.5+Hybrid neural networks	93.28	0.20
Hidden Markov model(HMM)	79.00	-

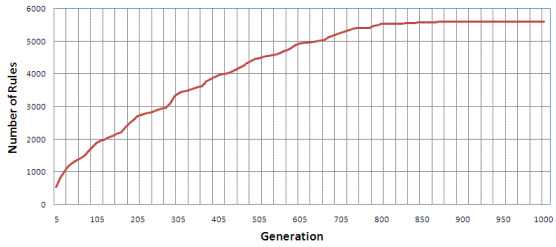

Fig. 10. **Number of Extracted Rules in Simulation.2**

and teardrop are considered unknown. By using the classifier we proposed to label the testing data, we obtained the testing results with different *k* values shown in Table.6 and Table.7, where *T* represents testing label and *R* represents real label.

TABLE VII
TESTING RESULTS OF THE PROPOSED METHOD WITH $k = 0.5$ FOR ANOMALY DETECTION

	Normal(T)	Intrusion(T)	Total
Normal(R)	174	20	194
Intrusion(R)	12	567	579
Total	186	587	773

TABLE VIII
TESTING RESULTS OF THE PROPOSED METHOD WITH $k = 0.7$ FOR ANOMALY DETECTION

	Normal(T)	Intrusion(T)	Total
Normal(R)	180	14	194
Intrusion(R)	29	550	579
Total	209	564	773

$$DR = (174 + 567)/773 = 95.9\% \quad (10)$$

$$PFR = 20/194 = 10.3\% \quad (11)$$

$$NFR = 12/579 = 2.1\% \quad (12)$$

We can see from Table.6 and Eq.(10)-Eq.(12) that the proposed method resulted in very high detection rate even the intrusion is unknown and very low negative false rate, which means there are scarcely cases that intrusion will be treated as normal. However, the trade-off is fairly high positive false rate.

$$DR = (180 + 550)/773 = 94.4\% \quad (13)$$

$$PFR = 14/194 = 7.2\% \quad (14)$$

$$NFR = 29/579 = 5.0\% \quad (15)$$

Table.7 and Eq.(13)-Eq.(15) suggest that the detection rate can remain high by adjusting *k* value to find the balance between PFR and NFR. Compared with other methods such as using GA and GP for anomaly detection mentioned in [11][12], for example, the GP method proposed by W. Lu and I. Traore[12] provides the detection rate around 57.14%, while our method can reach a higher detection rate 94.4% and a reasonable positive false rate. The most important advantage of our method is that no pre-experienced knowledge is needed.

C. Simulation.3 for the effectiveness of control parameters w_1 and w_2

The fitness value of the individual in this paper is defined as Eq.(3). Taking misuse detection as an example, we adjust the control parameters w_1 and w_2 to observe the effectiveness of these two parameters. We obtained the following results.

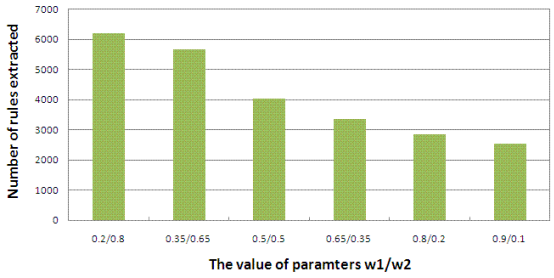

Fig. 11. **Number of Extracted Rules after 1000 generation with different control parameters**

From Fig.11, we find out that the higher parameter w_1 is, which means the lower parameter w_2 is, the number of rules extracted after 1000 generation is less, which shows that these two parameters can control the number of rules extracted from the training dataset.

On the other hand, we can conclude from Fig.12 that large number of rules don't always contribute to higher detection rate, for rules extracted in the rule pool may contain 'bad rules' which affect the detection rate. By adjusting the control

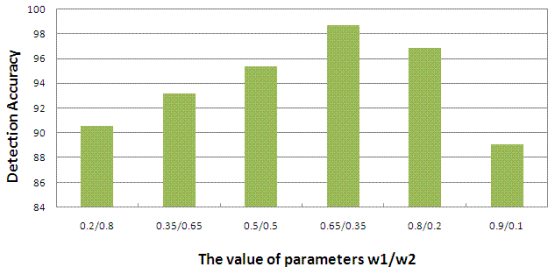

Fig. 12. **Detection Accuracy with different control parameters**

parameters w_1 and w_2, balance between the variety of rules (number of rules) and the high quality of rules (contribute to high detection rate) can be found. Here, the detection rate for misuse detection is as high as 98.7% when $w_1 = 0.65$ and $w_2 = 0.35$.

D. Comparison of the detection accuracy with crisp data mining and fuzzy data mining method

To examine the effectiveness of fuzzy mining method in network intrusion detection, we also conducted the simulation.1 and simulation.2 without fuzzy mining method, which means, continuous attributes are divided into two sub-attributes (the attribute greater than or equal to a threshold value and the attribute smaller than a threshold value). For each continuous attribute A_i, the mean value μ_i and standard deviation σ_i are calculated to obtain the initial threshold a_i selected randomly from $[\mu_i - \sigma_i, \mu_i + \sigma_i]$. Additionally, the initial threshold a_i is evolved by mutation in every generation so as to obtain more association rules. The evolution of thresholds is controlled by an additional mutation probability P_r, which is set at 1/3 in this paper.

Comparison of the detection accuracy for misuse detection and anomaly detection with crisp data mining and fuzzy data mining are shown respectively in the Table.8. and Table.9.

From Table.8 and Table.9, we can concluded that fuzzy data mining method contributes to both increasing detection rate and decreasing positive false rate and negative false rate in network intrusion problem. The reason fuzzy mining method outperforms the crisp data mining method is its characteristic of overcoming sharp boundary problem. Fuzzy sets can help to overcome this by allowing a partial membership to more than one set. Therefore, objects can be the members of more than one set and give a more realistic view on the data.

TABLE IX
COMPARISON OF THE DETECTION ACCURACY FOR MISUSE DETECTION WITH CRISP DATA MINING AND FUZZY DATA MINING

	Crisp data mining	fuzzy data mining
Detection Rate	98.3%	98.7%
Positive False Rate	0.67%	0.53%
Negative False Rate	5.0%	3.75%

TABLE X
COMPARISON OF THE DETECTION ACCURACY FOR ANOMALY DETECTION(K=0.7) WITH CRISP DATA MINING AND FUZZY DATA MINING

	Crisp data mining	fuzzy data mining
Detection Rate	90.3%	94.4%
Positive False Rate	10.3%	7.2%
Negative False Rate	9.5%	5.0%

VI. CONCLUSION

In this paper, we have proposed a fuzzy GNP-based Class Association Rule Mining with Sub-Attribute Utilization and Its Application to Classification, which can deal with discrete and continuous attributes at the same time and keep the completeness of data information. In addition, we applied them to both misuse detection and anomaly detection and did experiments with practical data provided by KDD99Cup and DAPRA98. The experiment results show that for misuse detection, the proposed method can provide high detection rate and low positive false rate, which is two important criteria for security systems; for anomaly detection, the method provide high detection rate and reasonable positive false rate even without pre-experienced knowledge, which is an important advantage over other methods.

REFERENCES

[1] K. Shimada, K. Hirasawa and J. Hu, "Genetic Network Programming with Acquisition Mechanisms of Association Rules", *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 10, No. 1, pp.102-111, 2006.
[2] T. Eguchi, K. Hirasawa, J. Hu and N. Ota, "A study of Evolutionary Multiagent Models Based on Symbiosis", *IEEE Trans. on Syst., Man and Cybernetics- Part B-*, Vol.36, No.1, pp.179-193, 2006.
[3] S. Mabu, K. Hirasawa and J. Hu, "A Graph-Based Evolutionary Algorithm: Genetic Network Programming(GNP) and Its Extension Using Reinforcement Learning", *Evolutionary Computation, MIT press*, Vol.15, No.3, pp.369-398, 2007.
[4] K. Hirasawa, M. Okubo, H. Katagiri, J. Hu and J. Murata, "Comparison between Genetic Network Programming (GNP) and Genetic Programming(GP)", *In Proc. of the Congress of Evolutionary Computation*, pp.1276-1282, 2001.
[5] K. Hirasawa, T. Eguchi, J. Zhou, L. Yu and S. Markon, "A Double-Deck Elevator Group Supervisory Control System Using Genetic Network Programming", *IEEE Trans. on Systems, Man and Cybernetics, Part C*, Vol.38, No.4, pp.535-550, 2008.
[6] K. Shimada, K. Hirasawa and J. Hu, "'Class Association Rule Mining with Chi-Squared Test Using Genetic Network Programming", *In Proc. of the IEEE SMC 2006, Taipei*, pp.5338-5344, 2006.
[7] W. Lee and S. J. Stolfo, "A Framework for Construction Features and Models for Intrusion Detection System"', *ACM Transactions on Information and System security*, Vol. 3, No. 4, pp.227-261, November 2000.
[8] Tcptrace software tool, www.tcptrace.org.
[9] W. Lee and S. J. Stolfo, "Data Mining Approaches for Intrusion Detection", *In Proc. of the 1998 USENIX Security Symposium,1998.*
[10] DARPA data, http://www.ll.mit.edu.
[11] M. Crosbie and G. Spafford, "Applying genetic programming to intrusion detection", *Technical Reportk, FS-95-01, AAAI Fall Symposium Series, AAAI Press, 1995.*
[12] W. Lu and I. Traore, "Detecting new forms of network intrusion using genetic programming", *Computational Intelligence*, Vol. 20, No. 3, pp.474-494, 2004.