An Efficient Model for Network Intrusion Detection System based on an Evolutionary Computational Intelligence Approach

T.AnithaDevi
PG Student
P.S.R Engineering College
Sivakasi

K.Ruba soundar
Professor/Dept. of CSE
P.S.R Engineering College
Sivakasi

ABSTRACT
Intrusion Detection systems are increasingly a key part of system defence. Various approaches to Intrusion Detection are currently being used but false alarm rate is higher in those approaches. Network Intrusion Detection involves differentiating the attacks like DOS, U2L, R2L and Probe from the Normal user on the internet. Due to the variety of network behaviors and the rapid development of attack fashions, it's necessary to develop an efficient model to detect all kinds of attacks. Building an effective IDS is an enormous knowledge engineering task. Characteristics of computational intelligence systems such as adaptation, fault tolerance, high computational speed and error resilience in the face of noisy information fit the requirements of building a good intrusion model. In this paper, we propose a network intrusion detection model based on evolutionary optimization technique called Genetic Network Programming (GNP) with sub attribute utilization mechanism. The proposed model is evaluated using KDDCup99 Dataset for misuse detection and using DARPA 98 Dataset for anomaly detection, which shows higher detection rate as well as low false alarm rate.

General Terms
Network Security, Intrusion Detection System, Computational Intelligence.

Keywords
Anomaly detection, fuzzy data mining, Genetic Network Programming**,** Misuse detection.

1. INTRODUCTION
In recent years, as internet and personal computers are populated, utilization rate of internet keeps increasing. It is changing people's lives gradually, and the majorities of people study, recreate, communicate and buy through internet. Besides common people, enterprise structure and business mode also undergoes transformation due to internet, and large enterprise or government organizations, in order to achieve operation purpose and efficiency, develop many application and service items resting on internet; these are an irresistible tendency in the new era. However, though internet brings about convenience and real timeliness, consequently comes information safety problem. As a result, network intrusion detection system (NIDS) has become an indispensable component of security infrastructure to detect these threats before they inflict widespread damage.

Network intrusion detection is the problem of detecting unauthorised use of, or access to, computer system over a network. A basic premise for intrusion detection is that when audit mechanisms are enabled to record system events, distinct evidence of legitimate activities and intrusion will be manifested in the audit data.

Computational intelligence (CI) is the study of the design of intelligent agents. An intelligent agent is a system that acts intelligently. It is flexible to changing environments and changing goals, it learns from experience, and it makes appropriate choices given perceptual limitations and finite computation.

1.1 Taxonomy of IDS
In general IDS fall into two categories according to the detection methods they employ, namely misuse detection and anomaly detection [1]. Misuse detection identifies intrusions by matching observed data with pre-defined descriptions of intrusive behaviour. Anomaly detection builds models for normal behaviour and detects anomaly in observed data by noticing deviations from the models.

As far as the data source is concerned, IDS can be classified into Host based and network based detections [2]. Host-based approaches detect intrusions utilizing audit data that are collected from the target host machine. Network-based approaches detect intrusions using the IP package information collected by the network hardware such as routers and switches.

1.2 Genetic Network Programming
GNP is one of the evolutionary optimization techniques, which uses the directed graph structure as genes instead of strings in genetic algorithm or trees in genetic programming. GNP is applied to dynamic problems based on inherent features of the graph structure such as reusability of nodes.

The basic structure of GNP is shown in the Fig.1. GNP comprises of three types of nodes Starting node, judgement node and processing node. Starting node initiates the GNP operations. Judgement nodes performs as a decision making function. Processing nodes act as a processing function.

Three kinds of genetic operators, i.e., selection, mutation, and crossover, are implemented in GNP [12].

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 Pc.

3) Mutation: One new individual is generated from one original individual by the following operators. 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. Fig.4. depicts the general gene structure of GNP individual.

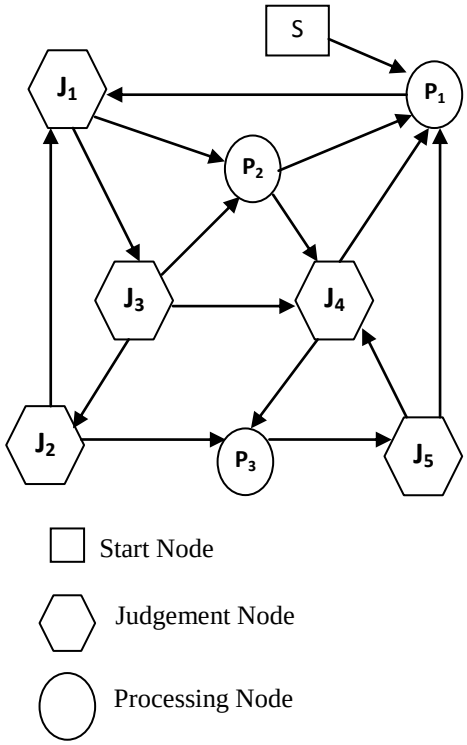

Fig 1: Basic structure of GNP

NT_i	ID_i	C_{i1}	C_{i2}	C_{i3}	

0	0	0	6			
1	1	1	3	7		
2	1	2	1	8		
3	1	3	2	4	7	
4	1	4	6	7	8	
5	1	5	4	6		
6	2	1	1			
7	2	2	6			
8	2	3	5			

Fig 2: Gene structure of GNP

2. RELATED WORKS

In the related works, different kinds of computational intelligence techniques used for NIDS modeling are discussed.

2.1 Artificial Neural Networks (ANN)

ANN consists of a collection of processing units called neurons that are highly interconnected in a given topology. ANNs have the ability of learning–by-example. This technique categorized into two, Supervised learning and Unsupervised learning. For supervised learning for intrusion detection, there are mainly supervised neural network (NN)-based approaches, and support vector machine (SVM)-based approaches. Bonifacio et al. [4] propose an NN for distinguishing between intrusions and normal behaviours. They unify the coding of categorical fields and the coding of character string fields in order to map the network data to an NN. Mill and Inoue [5] propose the Tree SVM and Array SVM for solving the problem of inefficiency of the sequential minimal optimization algorithm for the large set of training data in intrusion detection. Self-Organizing maps and adaptive resonance theory are two typical unsupervised neural networks. Hoglund et al. [6] extract features that describe network behaviours from audit data, and they use the SOM to detect intrusions.

2.2 Fuzzy Logic

Fuzzy logic, dealing with the vague and imprecise is appropriate for intrusion detection. Bridges et al. suggested to the use of fuzzy association rules and fuzzy sequential rules to mine normal pattern from audit data [7]. Flozer et al. [8] described an algorithm for computing the similarity between two fuzzy association rules based on prefix trees. Cho et al. [9] trained multiple HMMs were sent to the fuzzy inference engine, which gave a fuzzy normal or abnormal result.

2.3 Evolutionary Computation

Evolutionary Computation is a process gleaned from evolution in nature, is capable of addressing real-world problem with greater complexity. Wei LU el at. [10] propose a rule evolution approach based on Genetic Programming for detecting attack on the network.

2.4 Artificial Immune System (AIS)

AIS based intrusion detection systems perform anomaly detection. However instead of building models for the normal, they generate non-self patterns by giving normal data. Any matching to non-self patterns will be labeled as an anomaly [3].

3. PROPOSED FRAMEWORK

In the proposed model, fuzzy class-association rule mining method based on GNP is introduced. Association rule mining is used to discover association rule or correlations among a set of attributes in a dataset.

For misuse detection, the normal-pattern rules and intrusion-pattern rules are extracted from the training dataset. Classifiers are built up according to these extracted rules. While, for anomaly detection, we focus on extracting as many normal-pattern rules as possible. Extracted normal-pattern rules are used to detect novel or unknown intrusions by evaluating the deviation from the normal behaviour. The features of the proposed method are summarized as follows.

1) GNP-based fuzzy class-association-rule mining can deal with both discrete and continuous attributes in the database, which is practically useful for real network-related databases.

2) Sub attribute utilization considers all discrete and continuous attribute values as information, which contributes to avoid data loss and effective rule mining in GNP.

3) The proposed fitness function contributes to mining more new rules with higher accuracy.
4) The proposed framework for intrusion detection can be flexibly applied to both misuse and anomaly detection with specific designed classifiers.
5) Experienced knowledge on intrusion patterns is not required before the training.
6) High detection rates (DRs) are obtained in both misuse detection and anomaly detection.

3.1 Sub Attribute and Fuzzy Membership Function Construction

Network connection data have their own characteristics, such as discrete and continuous attributes, and these attribute values are important information that cannot be lost. A sub attribute-utilization mechanism concerning binary is introduced, 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. Each value of continuous attributes in the database is transformed into three linguistic terms (low, middle, and high). A predefined membership function is assigned to each continuous attribute and the linguistic terms can be expressed by the membership function.

The parameters α, β, and γ in a fuzzy membership function for attribute A_i is set as follows:

β = average value of attribute Ai in the database;
γ = the largest value of attribute Ai in the database;
$\alpha + \gamma = 2\beta$

3.2 GNP based Class-Association Rule Extraction Process and Updating Rule Pool

Fig. 2 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 judgment node and 2 for processing node. ID_i serves as an identification number of a judgment or processing node, for example, $NT_i = 1$ and $ID_i = 2$ represents node function J_2. C_{i1}, C_{i2}. . . denote the node numbers connected from node i. The total number of nodes in an individual remains the same during every generation.

The following is a statement of association-rule mining [3]. Let $I = \{A_1, A_2, ..., A_i\}$ be a set of literals, called items or attributes. Let G be a set of tuples, where each tuple T is a set of attributes such that $T \subseteq I$. Let TID be an ID number associated with each tuple. A tuple T contains X, a set of some attributes in I, if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \phi$. X is called antecedent and Y is called consequent of the rule. If the fraction of tuples 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 by support $(X \cup Y)$/support(X).

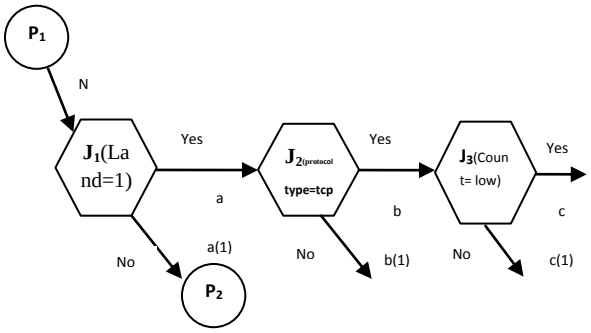

Fig 3: Node transition to find class-association rule

Let Ai be an attribute in a database with value 1 or 0, and k be class labels. Then, a class-association rule can be represented by as a special case of the association rule $X \Rightarrow Y$ with fixed consequent C.

$(A_p = 1) \wedge \cdots \wedge (A_q = 1) \Rightarrow (C = k) k \in \{0, 1\}$

A judgment node in GNP has a role in checking an attribute value in a tuple. Candidate class-association rules are represented by the connections of judgment nodes. An example of the representation is shown in Fig.3. 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

$(A_1=1) \Rightarrow (C=1)$
$(A_1=1) \wedge (A_2=1) \Rightarrow (C=1)$
$(A_1=1) \wedge (A_2=1) \wedge (A3=1) \Rightarrow (C=1)$
$(A_1=1) \Rightarrow (C=0)$
$(A_1=1) \wedge (A_2=1) \Rightarrow (C=0)$
$(A_1=1) \wedge (A_2=1) \wedge (A_3=1) \Rightarrow (C=0)$

are examined by the node transition in Fig.5. 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, J2...) equals the number of attributes (A_1, A2...) in the database.

In Fig. 3, N is the total number of tuples. a, b, and c are the numbers of tuples moving to Yes-side at judgment nodes J_1,J_2 and J_3 respectively. a(1),b(1) and c(1) are the numbers of tuples in class 1 moving to Yes-side at the judgment nodes, respectively. Actually, the processing node from which the node transition starts saves the counted numbers and calculates the measurements. For example, in the case of the rule $(A_1=1) \Rightarrow (C=1)$, the support is a(1)/N and the confidence is a(1)/a. In the case of $(A_1=1) \wedge (A_2=1) \wedge (A_3=1) \Rightarrow (C=1)$,

the support is c (1)/N and the confidence is c (1)/c. $\chi 2$ values are also calculated.

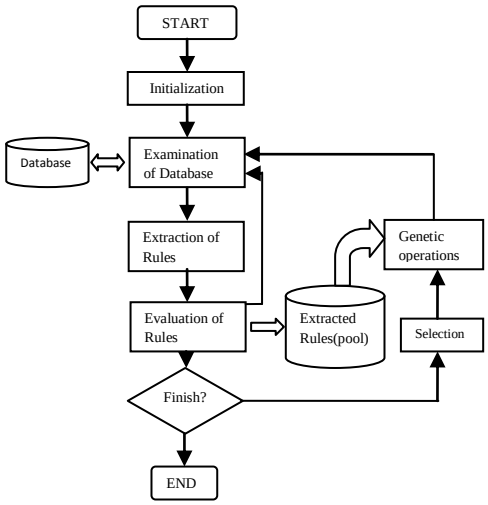

Fig 4: Flow diagram for rule extraction process

The distinguished feature of GNP is reusability of nodes based on the graph structure, so the nodes used in a certain transition can be shared with other node transitions. As a result, many kinds of association rules can be extracted by compact structures. In addition, the extracted rules that satisfy the conditions of the measurements are stored in an association rule pool every generation. Therefore, the rule extraction of GNP is carried out throughout the generations in order to create the rule pool with sufficient rules. In other words, the aim of GNP is not to create the optimal individual that can extract good rules, but to extract many good rules by making use of all the individuals in all the generations an store the rules in the pool.

GNP examines the attributes of tuples at judgment nodes and calculates the measurements of association rules at processing nodes. 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. The total number of tuples moving to Yes-side at each judgment node is memorized in the processing node from which the node transition rule extraction starts.

The extracted fuzzy class-association rules are stored in a fuzzy rule pool. When an important rule is extracted by GNP, it is stored in the pool with its support, confidence, $\chi 2$ value, and the parameters of the fuzzy membership function. Occasionally, a fuzzy rule already stored in the pool would be extracted again. In that case, the membership function and $\chi 2$ value might be changed. If the fuzzy rule has higher $\chi 2$ value, it will replace the same old fuzzy rule in the pool along with its fuzzy parameters. Therefore, the pool is updated every generation and only important fuzzy rules with higher $\chi 2$ values and better-adapted fuzzy parameters are stored.

3.3 Misuse Detection Classifier

The classifier for misuse detection calculates the average matching between new connection data dnew and all the rules in the normal rule pool, i.e., $MATCH_n$ (dnew), and the average matching between new connection data dnew and all the rules in the intrusion rule pool, i.e., MATCHi (dnew). If $MATCH_n$ (dnew) $\geq$ $MATCH_i$ (dnew), new connection data dnew is labeled as normal. If $MATCH_n$ (dnew) < $MATCH_i$ (dnew), new data dnew is labeled as intrusion. Fig.5. gives an example of the misuse detection classifier.

3.4 Anomaly Detection Classifier

First, the matching between each training data d of the normal connections and the rules in the normal rule pool, i.e. $MATCH_n$ (d), is calculated. Then, mean μ and standard deviation σ of $MATCH_n$ (d) of all the training data are calculated. In the testing period, when a new unlabeled connection data comes, the matching between the new data dnew and the rules in the normal rule pool is calculated. If MATCHn (dnew) < $(\mu - k\sigma)$, the connection is labeled as intrusion. If MATCHn (dnew) $\geq$ $(\mu - k\sigma)$, it is labeled as normal. By adjusting parameter k, balance the PFR and negative false rate (NFR).

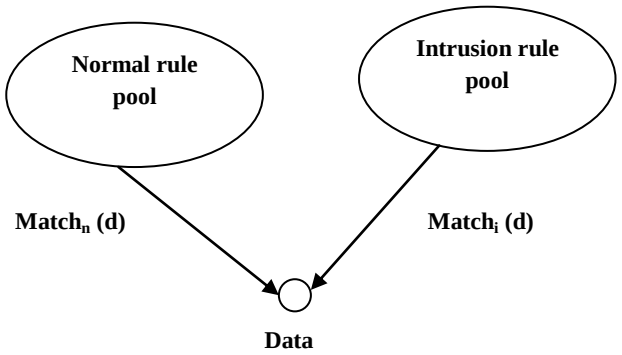

Fig 5: Example of misuse detection classifier

4. SIMULATIONS

The proposed method for misuse detection is carried out with KDD99Cup database in order to compare with other machine learning methods. The training dataset contains 3342 connections randomly selected from KDD99Cup database, among which 1705 connections are normal and the other 1637 connections are intrusion, where three types of attacks (nepture, smurf, and portsweep) are included. A total of 41 attributes are included in each connection; however, after the attribute division, 113 sub attributes are assigned to the judgment functions in GNP.

Table 1. Simulation Parameters for GNP

Population size	120
Generation	1000
Processing node	10
Judgement node	100
Crossover Rate	1/5
Mutation Rate	1/3

4.1 Fitness and Genetic Operation

Each obtained rule is checked by the training data to get the fitness value. The scale of the fitness value is [-1, 1]. Higher fitness of a rule results in high DR and low positive false rate (PFR), which means the rate of incorrectly assigning normal connections to a intrusion class. On the other hand, lower fitness results in low DR and high PFR. When a rule is

extracted by GNP, the overlap of the attributes between the rule and the already stored rules is checked to confirm whether the rule is newly extracted or not.

The fitness of an individual, the fitness of extracted rule r is defined as follows

$$fitness_r = \frac{Nt_c}{Nt} - \frac{Nn_i}{Nn} \quad (1)$$

Where

Nt_c , the number of connections correctly detected by rule r;

Nt , the number of connections in the training data;

Nn_i , the number of normal connections incorrectly detected by rule r;

Nn , the number of normal connections in the training data.

The fitness of a GNP individual for network intrusion problems is defined by

$$F = \sum_{r \in R} \{w_1 * fitness + w_2 * \alpha_{new}(r)\} \quad (2)$$

Where

R, set of suffixes of association rules extracted by the individual;

$\alpha_{new}(r) = \alpha_{new}$, if rule r is new

0, otherwise

W_1, W_2, control parameters.

Every generation, individuals are replaced with the new ones by the genetic operators namely selection, crossover and mutation in order to obtain more class-association rules.

4.2 Misuse Detection

The testing database contains 750 unlabeled normal connections and 240 unlabeled intrusion connections. The detection results obtained by the proposed misuse detection classifier are shown in Table 2, where T represents the label of the testing results given by the classifier and C represents the correct label. Three criteria are used to evaluate our testing results, i.e., DR, PFR, and NFR. DR means the total DR, PFR means the rate at which the normal data are labelled as intrusion, and NFR means the rate at which the intrusion data are labelled as normal.

Table 2. Testing Results for Misuse Detection

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

DR = (746 + 231)/990 = 98.7% (3)
PFR = 4/750 = 0.53% (4)
NFR = 9/240 = 3.75% (5)

Compared with the results obtained by other machine-learning techniques dealing with KDD99Cup, it is found that the proposed method for misuse detection provides higher DR than most of the machine-learning techniques except the combination method of support vector machine (SVM) with GA and SVM with fuzzy logic.

4.3 Anomaly Detection

The proposed method for anomaly detection is evaluated by the simulations with DARPA98 database. The training database is intrusion free for the purpose of the anomaly detection. It contains 9137 normal connection records. After preprocessing, 30 attributes are included in every connection record. However, after the attribute division, 82 sub attributes are assigned to the judgment functions in GNP. After 1000 generations, 5589 rules related to the normal connections are extracted. The testing database contains 773 connection records including 194 unlabeled normal records and 579 unlabeled intrusion records. Because the training database is intrusion-free, all kinds of intrusions such as back, ipsweep, land, neptune, pod, port sweep, satan, smurf, and teardrop are considered unknown. After the classification using the proposed anomaly detection classifier, the testing results under different settings of k are obtained, as shown in Tables 3 and 4.

Table 3. Testing Results for Anomaly Detection with K=0.5

	Normal (T)	Intrusion (T)	Total
Normal (C)	150	44	194
Intrusion (C)	4	575	576
Total	154	619	775

DR, PFR, and NFR in the case of k = 0.5 are

DR = (174 + 567)/773 = 95.9% (6)
PFR = 20/194 = 10.3% (7)
NFR = 12/579 = 2.1% (8)

From Table 3, the high DR and low NFR are obtained even if the intrusion is unknown, which means that the intrusion will be treated as normal with low probability. However, the difficult point is the tradeoff between PFR and NFR because PFR is high in this case.

Table 4. Testing Results for Anomaly Detection with K=0.7

	Normal (T)	Intrusion (T)	Total
Normal (C)	174	20	194
Intrusion (C)	55	524	579
Total	229	544	773

In the case of k = 0.7,

DR = (180 + 550)/773 = 94.4% (9)
PFR = 14/194 =7.2% (10)
NFR = 29/579 = 5.0% (11)

Table 4 suggests that the DR still remains high even after adjusting k to find the balance between PFR and NFR. Compared with the method using GP for anomaly detection [10], which provides the DR around 57.14%, the proposed method can reach a higher DR 94.4% and a reasonable PFR. The most important advantage of our method is that no pre experienced knowledge is needed. The proposed method works well without detailed knowledge on the network intrusion such as intrusion types. Even without pre experienced knowledge, the proposed method still provides higher DR, which indicates GNP could be a potentially effective algorithm for anomaly detection.

The Fig.6 & 7 shows the comparison of intrusion detection rate and false alarm rate for the proposed framework with various approaches.

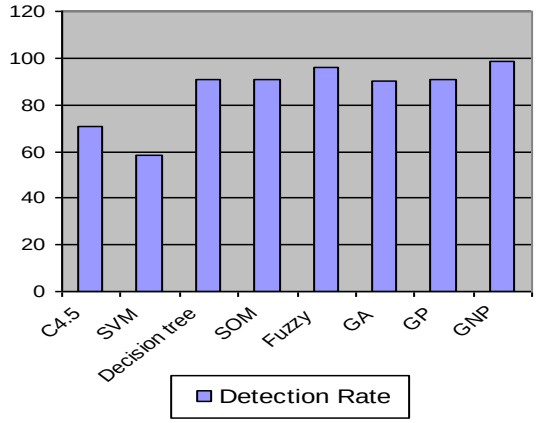

Fig 6: Comparison of Detection Rate for various approaches

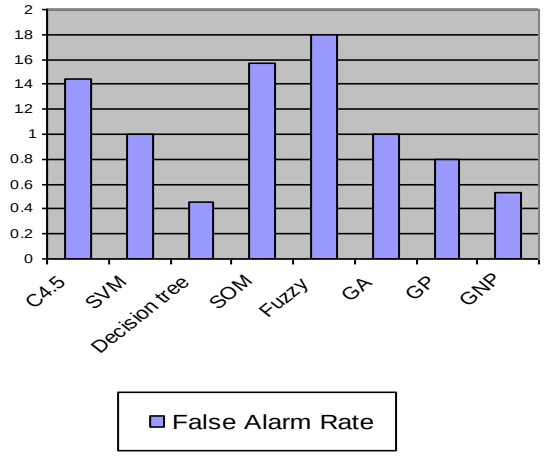

Fig 7: Comparison of False alarm Rate for various approaches

5. CONCULSION

In this paper, intrusion-detection classifiers for both misuse detection and anomaly detection have been developed using a GNP-based fuzzy class-association-rule mining with sub attribute utilization and their effectiveness is confirmed using KDD99Cup and DARPA98 data. The simulation results in the misuse detection show that the proposed method shows high DR and low PFR, which are two important criteria for security systems. In the anomaly detection, the results show high DR and reasonable PFR even without pre experienced knowledge, which is an important advantage of the proposed method.

6. FUTURE WORK

In the future, we will focus on building distributions (probability density functions) of normal and intrusion accesses based on fuzzy GNP. By using the probability distributions, the data can be classified into normal class, known intrusion class and unknown intrusion class. In addition, the new data (testing data) can be labeled as normal or intrusion with a certain probability by using the distributions**.**

7. REFERENCES

[1] H.Debar, M.Dacier, A.Wespi, "Towards a taxonomy of intrusion-detection systems", Computer Networks 31 (8) (1999) 805-822

[2] S.Chebrolu, A.Abraham, and J.P.Thomas, "Feature deduction and ensemble design of intrusion detection systems", Comput.Secur. vol.24, no.4,pp, 295-307, Jun. 2005.

[3] K. Shimada, K. Hirasawa, and J. Hu, "Class association rule mining with chi-squared test using genetic network programming," in Proc. IEEE Int. Conf. Syst., Man, Cybern., 2006, pp. 5338–5344.

[4] J. M. Bonifacio, Jr., A. M. Cansian, A. C. P. L. F. De Carvalho, and E. S. Moreira, "Neural networks applied in intrusion detection systems," in Proc. IEEE Int. Joint Conf. Neural Netw., 1998, vol. 1, pp. 205–210.

[5] J. Mill and A. Inoue, "Support vector classifiers and network intrusion detection," in Proc. Int. Conf. Fuzzy Syst., 2004, vol. 1, pp. 407–410.

[6] J. Hoglund, K. Hatonen, and A. S. Sorvari, "A computer host based user anomaly detection system using the self-organizing map," in Proc. Int. Joint Conf. Neural Netw., 2000, vol. 5, pp. 411–416..

[7] S.M.Bridges, R.B. Vaughn, Fuzzy data mining and genetic algorithms applied to intrusion detection in: Proceedings of the 23rd National Information Systems Security Conference, 2000, pp.13 -31.

[8] G.Florez, S.M. Bridges, R.B. Vaughn, An improved algorithm for fuzzy data mining for intrusion detection in: Proceedings of the 21st International Conference of the NAFIPS'02, pp, 457-462.

[9] S-B, Cho, Incorporating soft Computing techniques into a probabilistic intrusion detection system, IEEE transactions on Systems, Man and Cybernetics: Part C: Applications and Reviews 32 (2) (2002) 154-160.

[10] Wei Lu and Issa Traore, "Detecting new forms of network intrusion using genetic programming". Journal of Computational Intelligence, volume 20.(2004).

[11] Shelly Xiaonan, Wolfgang Banzhaf, "The use of computational intelligence in intrusion detection systems: A review", Science Direct-Applied Soft Computing 10 (2010) 1–35.

[12] S. Mabu, K. Hirasawa, and J. Hu, "A graph-based evolutionary algorithm: Genetic network programming (GNP) and its extension using reinforcement learning," Evol. Comput., vol. 15, no. 3, pp. 369–398, 2007.