Waseda University Doctoral Dissertation

Study on Intrusion Detection using Average Matching Degree Space based on Class Association Rule Mining

Nannan LU

Graduate School of Information, Production and Systems

Waseda University

January 2013

Abstract

With the increasing number of users and systems connected to networks, both Internet and individual systems are in danger of intrusion. Various intrusion preventions techniques have been implemented to protect computer systems in the form of authentication and firewalls. However, only the intrusion prevention is not sufficient, as those systems become more complex with the rapid growth and expansion of Internet technology and local network systems. Therefore, Intrusion Detection Systems are designed to keep computer systems security by monitoring Internet and individual systems for suspicious activities. The main intrusion detection techniques have misuse detection and anomaly detection. Misuse detection uses patterns of well known attacks to identify known intrusion activities. Whereas, it cannot detect unknown attacks without any pre-collected patterns. On the other hand, anomaly detection relies on the normal patterns to identify anomalies which deviate significantly from the normal patterns. But, it results in more false alarms.

In order to utilize the advantages of both misuse detection and anomaly detection, a hybrid framework of the intrusion detection system is implemented to combine the advantages of both misuse detection and anomaly detection using a series of data mining approaches. Class association rules are inductively learned from network connections and used as the basis of an intrusion detection system. In this thesis, class association rules are extracted by Genetic Network Programming(GNP), which is one of evolutionary algorithms. As the quality of the class association rules is essential for classification, an efficient two-stage rule pruning method intend to reduce the redundant and irrelevant information in the large number of rules. In the first stage, an average matching degree-based method is applied to pre-prune the rules in order to improve the efficiency of Genetic Algorithm(GA).

In the second stage, GA is implemented to pick up the effective rules among the remaining rules in the first stage. Simultaneously, in order to solve the sharp boundary problem in continuous attributes, Fuzzy set theory is integrated into GNP to discover class association rules.

To construct effective intrusion detection systems, classification is another central aspect to be studied. In this thesis, two classification approaches are proposed except the basic classifier. Distance-based classification approach makes use of the numeric distance of the new connection to its closest neighbor points to pre-classify it as normal or intrusion, then supposes the anomaly centroids based on the known information of normal and misuse intrusions to distinguish it as normal, misuse or anomaly intrusion accurately. Next, the other classification approach is proposed by combing the clustering and Gaussian functions to get the accurate boundary of normal and misuse intrusion.

The effectiveness and advantages of the proposed algorithms have been objectively evaluated on KDD Cup 1999 and NSL-KDD data sets.

Acknowledgements

I would like to thank my advisor, Professor. Kotaro Hirasawa, for his constant guidance and encouragement. During my three years' study at Waseda University, I get tremendous advices and supports from him. Without his supports, I would have never finished this research. I would also like to thank Dr. Shingo Mabu for his extensive help and comments on my research and papers. I thank them from the bottom of my heart.

I also would like to thank Prof. Furuzuki, Prof. Yoshie and Prof. Fujimura of Waseda University, who make efforts to provide me lots of helpful comments and advices to my doctor thesis.

I am grateful for the support received from China Scholarship Council.

Finally, I would like to thank my family for their love and constant support, and all friends who helped me during my three years' study in Waseda University.

Contents

List of Figures

List of Tables

1

Introduction

Since computer virus was first described by Fred Cohen(1), various types of attacks proliferated on computers and Internet. Some of them are produced manually by attackers, which aims at stealing the personal information. Some of them are programmed by hackers, which aim at disrupting the Internet and infecting other systems automatically. On the other hand, the number of users and systems connected to the networks grew dramatically as connecting to networks became more much easier, and the development of e-commerce and e-government dramatically accelerated this process. Many individual persons and organizations prefer using Internet for routine works and business affairs(2). Thus, security problems become serious in recent years.

Firewalls as a passive defense device is not enough to keep networks secure. Therefore, different kinds of Intrusion Detection Systems(IDSs) are designed to protect computer networks against various attacks. An excellent IDS has some inherent requirements. Its prime principle is to detect as many attacks as possible with minimum number of false alarms. In addition, it is also important to make the system adaptable and extensible.

1.1 Problems

Nowadays, computer systems are vulnerable to both abuse by insiders and penetration by outsiders. As defined by the SysAdmin, Audit, Network and Security (SANS) institute, intrusion detection is the act of detecting actions that attempt to compromise the confidentiality, integrity or availability of a resource(3). As a new and retrofit approach, intrusion detection becomes a challenging task since the increased connectivity

of computer systems gives greater access to outsiders and makes it easier for intruders to avoid identification(4).

On one hand, intrusions can attack the systems within a short time and infect other systems by Internet quickly. Besides, attackers often come up with newer and more advanced methods to defeat the installed security systems. In 1988, approximate 5,000 computers through the Internet were rendered unusable within 4 hours by a program called a worm. In 1993, more users of computer systems were alerted to such dangers when a set of programs called sniffers was placed on many computers run by network service providers and recorded login names and passwords(5).

On the other hand, the number of host users is growing and the number of corresponding connections to Internet is also increasing remarkably. According to the Internet System Consortium(ISC) survey, the number of hosts on the internet around 900,000,000 in Jan 2012 as Fig. 1.1 shows(6). Correspondingly, P. Lyman et.al. reported the estimated the size of the Internet to be 532,897TB by 2003(7). More and more business and individual affairs have been done on Internet. Therefore, an effective

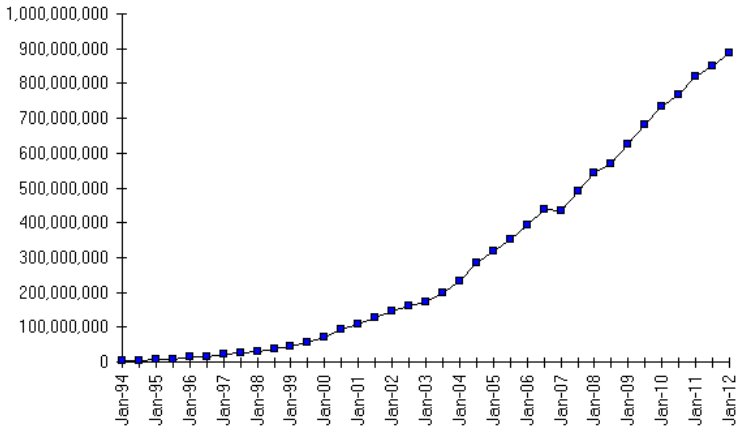


Figure 1.1: Internet Domain Survey Host Count

intrusion detection system is urgently needed for everyone.

Current intrusions can be categorized into diverse types. It is a challenge for an intrusion detection system to detect a wide range of intrusions with few false alarms. Hundred percent detection and no false alarms are ideal for an intrusion detection system. It becomes critical to build reliable intrusion detection systems which can

detect different types of attacks with very few false alarms in realistic environments. Therefore, the researches of this thesis aim at building a highly extensible and robust intrusion detection system.

1.2 Intrusion Detection and Intrusion Detection Systems

Intrusions can be generally distinguished into known intrusions and unknown intrusions. Both two kinds of intrusions may compromise confidentiality, integrity or availability of the systems or computers(8)(9)(3)(10). Therefore, broadly speaking, there are two kinds of intrusion detection techniques corresponding to known intrusions and unknown intrusions: misuse detection and anomaly detection(4)(11)(12).

Misuse detection(4)(13) essentially identifies the previously known attacks from normal network connection data. It utilizes the signatures of the known attacks and matches them against the observed activity. If it matches a previously known attack signature, the activity will be detected as an attack. However, if a new attack is produced, the system fails to recognize it. In conclusion, the main advantage of misuse detection is that it focuses on analyzing the known attacks and produces few false alarms. The main disadvantage of misuse detection is that it can detect only known attacks which have defined signatures.

Anomaly detection technique(4)(14)(15) establishes the profile of the normal activities of the computer or Internet. It looks for the deviation between the observed activity and normal patterns. Once it does, the observed activity is identified as anomaly intrusion. The main advantage of anomaly detection systems is that they can detect previously unknown attacks. By defining what's normal, they can identify the abnormal whether it is an attack or not. In actual systems, however, it results in a large number of false alarms. Anomaly detection systems are also difficult to be realized in highly dynamic environments.

There are two types of intrusion detection systems that employ one or both of the intrusion detection techniques introduced above(4). The principles of the Host-based IDS and Network-based IDS are very similar in that intrusion detection is based on analyzing the observed events for patterns, but their operations are quite different(16). Host-based systems(17)(18)(19) focus their analysis on user activity or program behavior at the operating system or application level, while network-based intrusion detection

systems obtain data by monitoring the traffic and examining network packets in the network to which the hosts are connected(20)(21).

Host-based intrusion detection systems(22) detect intrusions using audit data which are collected from the target host machine. As the information provided by the audit data can be extremely comprehensive and elaborate, host-based systems can obtain high detection rates and few false alarms. However, there are disadvantages for host-based approaches. Firstly, host-based systems cannot easily prevent attacks: when an intrusion is detected, the attack has partially occurred. Secondly, audit data may be altered by attackers, which influencs the reliability of audit data.

With the development of networks, more and more individual hosts are connected into local area networks or wide area networks. However, the hosts, as well as the networks, are exposed to intrusions due to the vulnerabilities of network devices and network protocols. The TCP/IP protocol can be also exploited by network intrusions such as IP spoofing, port scanning and so on. Therefore, network-based systems(23)(24)(25) detect intrusions using IP package information collected by the network hardware. More importantly, this type of systems can protect the host machine away from attacks, as their detection occurs before the data arrives at the machine. And it also can detect the attacks missed by host-based intrusion detection systems.

1.3 Literature Review

Intrusion detection(4)(26)(27) has been developing for over 20 years since D. E. Denning first proposed intrusion detection as a different notion of security in computer system(28). In this historical process, the researchers have proposed and implemented various techniques(29)(30)(20)(31) in this field. Early works on intrusion detection was due to Anderson(32) and Denning(28). Since then, it has become a very active field. There has been steadily growing interest in research and development of IDS. The main goal was to create a system capable of detecting different kinds of attacks. To accomplish this goal, researchers have been exploring various approaches such as Pattern Matching(33)(34), Statistical Models(35)(36), Information Theoretic measures(37), Data Mining(38)(39), Immune System(40) and Machine Learning(41)(42)(43) etc.. The intent of the followings is to give a brief overview of recent intrusion detection approaches on some of these fields.

1.3.1 Statistical Modeling

Statistical Modeling(35)(36)(44)(45) is one of the earliest methods used for anomaly detection. It measures the user and system behavior by a number of variables sampled over time, and builds profiles based on the variables of normal behaviors. The actual variables are then compared against the profiles, and deviations are considered abnormal.

There are many statistical techniques(46). Denning(28) proposed a statistics models for intrusion detection. According to audit data, the variables were represented as different metrics. Then, to describe the profiles of variables, a serious of statistical models were built, including mean and standard deviation, multivariate model, Markov process model and time series model. But, these methods construct too simple models leading to worse discrimination. The next generation intrusion detection expert system is the representative IDS based on statistics, which measures the similarity between long-term behaviors and short-term behaviors of the systems for intrusion detection(47). (48) examined the application of Statistical Traffic Modeling for detecting novel attacks against computer networks. In this method, Kolmogorov-Smirnov statistics was used to model and detect DoS as well as probing attacks. Ye et.al.(49) developed an anomaly detection technique, where the norm profile of temporal behaviors learns the Markov chain model from computer connection data, and detects anomalies based on the Markov chain model of temporal behaviors. However, this method could not provide accurate classification since various features of the connection level are ignored. The multiple linear regression analysis which is one of multivariate statistical analysis methods was used in (50) to analyze network traffic. But, it is not suitable to express an attack using the linear model since it may be nonlinear.

In short, these approaches in anomaly detection require the construction of a model for normal user behaviour, and any user behaviour that deviates significantly from this normal behaviour is flagged as an intrusion. It can also be difficult to determine the correct anomaly threshold at which behaviour is to be considered an intrusion. Also, to apply statistical techniques, too many assumption conditions are needed, which may contradict the facts.

1.3.2 Supervised Learning Approaches

Supervised learning is the machine learning task of building a model using labeled training data.

Naive Bayes classifier(51) is composed of Directed Acyclic Graph(DAG) which is trained as well as Conditional Probability Table(CPT) by the training connection data. Then, it is possible to classify any new data with its attributes' values using the Bayes rules based on the quantified network structure. However, Naive Bayes classifier makes a strong independence relation assumption between features when the features are correlated.

S. Mukkamal et.al(52) implemented Support Vector Machine(SVM) to classify new connection data into normal and intrusion by mapping real valued input feature vectors to a higher dimensional feature space. It has the advantage of dealing with high dimensionality of data. But, the performance of SVM approach lies in the choice of the kernel, which makes it difficult to deal with large scale database.

Neural networks are also used to realize IDS in many researches(53). They are algorithmic techniques(54)(55) which are used to first learn the relationship among information and then generalize to obtain new input-output pairs in a reasonable way. Multi Layer Perceptron(MLP) is a feed forward artificial neural network model that maps the set of input data into the set of appropriate outputs(56). A MLP consists of multiple layers of nodes in a directed graph, with each layer being connected fully to the next layer. In (57), MLP is the basic unite of the ensemble classifiers. In this way, the different sources of information are integrated with each other, which is called data fusion. Although the neural networks can work effectively with noisy data, they require a large amount of data for the training and it is often hard to select the best architecture for the neural networks. Adaboost is an important method of ensemble learning. It is a stereotype algorithm of boosting, whose basic idea is to select and combine a group of weak classifiers to form a strong classifier(58). But, a group of weak classifiers is required to be designed beforehand. In(20), weak classifier is constructed by the decision stump which is a decision tree with a root node and two leaf nodes. However, the performance of Adaboost algorithm always relies on the weak classifiers. In addition, it is easily influenced by noises.

1.3.3 Unsupervised learning Approaches

In machine learning, unsupervised learning refers to the problem of trying to find hidden structure in unlabeled data. Data clustering is a main type of unsupervised methods, such as K-means and fuzzy c-means(59)(60). One of the main drawbacks of the clustering technique is that it is based on calculating the numeric distance between the observations, hence the observations must be numeric. Observations with symbolic features cannot be easily used for clustering, which result in inaccuracy. In addition, the clustering methods consider the features independent and are unable to capture the relationship among different features of a single record, which further degrades the detection accuracy of the attacks.

A self-organizing map (SOM), also known as Kohonen map, is a typical unsupervised neural network based on competitive learning. It can organize and train the structure of neural networks by itself. Except input layer and output layer, it has a competitive layer. Hoglund et al.(61) extract features that describe network behaviors from audit data, and they use the SOM to detect intrusions. Kayacik et al.(62) propose a hierarchical SOM approach for intrusion detection. Specific attention is given to the hierarchical development of abstractions, which is sufficient to permit direct labeling of SOM nodes with connection type.

1.3.4 Data Mining Approaches

Data mining approaches generally discover relevant patterns of programs and user behaviors, which are mainly in the form of rules or frequent episodes. Lee et. al(63) develop a data mining framework MADAM ID (Mining Audit Data for Automated Models for Intrusion Detection) for mining audit data to discover useful frequent patterns and association rules. In this approach, the learned rules replace the manually encoded intrusion patterns and profiles. PIPPER, a rule learning tool, has been used for automatic construction of the detection models in MADAM ID. ADAM(Audit Data Analysis and Mining)(64) uses a frequent itemset-based association rule mining algorithm in detection as an online network-based IDS. The framework of ADAM has two phases: training phase and on-line phase. In the training phase, the attack-free training data is fed to a module whose output is a rule-based profile of normal activities. After

that, the produced profile is input to another module to perform a dynamic on-line algorithm with association rules.

Some approaches belonging to Soft Computing(SC) are used to find rules from either host audit data or network traffic. Generally speaking, SC is a methodology that provides flexible information processing capability for handling real-life ambiguous situations(65). Since its features of flexibility and adaptability in new environments and ability of generalizing from training data, it can be widely used in IDS.

Fuzzy set theory(54) is a mathematical technique for dealing with imprecise data and problems with many solutions. It can deal with continuous attributes and handle sharp boundary problems, whose advantages make the rules more comprehensible for humans. (66) proposed Fuzzy Intrusion Recognition Engine(FIRE), which is a network-based intrusion detection system that uses fuzzy system to assess malicious activity against computer networks. This system uses agent to perform its own fuzzification of input data sources. At the end, all agents communicate with a fuzzy evaluation engine that combines the results of individual agents using fuzzy rules to produce alerts that are true to a degree. A. Tajbakhsh et. al(67) discover fuzzy association rules to build the classifier. In this approach, a set of fuzzy association rules is extracted for each class. In order to determine the label of a new connection data, the similarity of the extracted rules to the new connection data is calculated.

As to Evolutionary Computation, Genetic Algorithm(GA)(68) is referred firstly. In IDSs, GA can be used to evolve simple rules for either host audit data or network traffic. Besides, GA is a good tool for feature selection(69)(70)(71)(70) or model selection(71). In this sense, GA is also used to find the suitable fuzzy membership functions(72). The final best set of fuzzy membership functions in all populations is gathered to be used for mining fuzzy association rules. However, most of rule mining approaches tend to produce a large number of rules that increase the complexity of the system. In order to solve this problem, many rule pruning methods are proposed. One well known research direction is to remove redundant rules using the concepts of closed item sets(73) and representative rules(74). In (75), an algorithm taking advantage of upward closure properties of weak rules is proposed to find a small subset of a class association rule set. And the clustering of association rules has been also used to obtain a reasonable set of rules(76). But, most of the algorithms need users' information on the complete rule set. Genetic Programming (GP) has been also applied to intrusion

detection. GP ensemble(77) applies cellular GP to create classifiers, which creates some independent decision trees based on different training data and they are finally combined to form the intrusion detection system. On the other hand, Genetic Network Programming (GNP)(78) has been proposed as an extension of Genetic Algorithm and Genetic Programming(GP). GNP-based data mining has been already applied to intrusion detection systems. The basic framework of IDS using GNP is described in (30) where fuzzy class association rule mining is developed and misuse detection and anomaly detection are realized. Then, Fuzzy set theory is induced into GNP to propose the Fuzzy GNP to extract class association rules and misuse detection and anomaly detection are integrated to propose a hybrid intrusion detection system(79).

1.4 Motivations and Contributions of the Thesis

By observing the drawbacks of other methods, data mining as descriptive method is still a safe and efficient method(74). The main theme of the proposed approach is to apply data mining algorithms to extract rules from data, which accurately capture the actual behavior of various intrusions and normal activities. From this view, the framework of the proposed intrusion detection systems mainly contain three parts: rule mining, rule pruning and classification.

1.4.1 Building a Hybrid Framework for Intrusion Detection System

Traditional misuse detection and anomaly detection have advantages and disadvantages. Misuse detection is effective to the intrusions seen previously. Whereas, anomaly detection is effective to the intrusions never seen before. In reality, a new produced connection can be normal, known intrusion or unknown intrusion. Therefore, a hybrid system is needed to detect known intrusions and unknown intrusions simultaneously by combining misuse detection and anomaly detection.

In this hybrid framework, a new matching measure is proposed to evaluate how much a data matches with the rules in different classes. It is named as average matching degree. Different from classical method using the highest confidential rule, each rule can contribute to the final classification. More importantly, the multi-dimensional problem can be projected into a two dimensional average matching degree space in the case of intrusion detection.

1.4.2 Using GNP to Extract Class Association Rules for Intrusion Detection

Association rule mining is one of the most popular methods in data mining. Different with association rules, the consequent parts of the class association rules are represented as class labels. Therefore, class association rules are much more appropriate to apply to classification problems than association rules. Both of them can be used to discover various of associations among the attributes. Furthermore, class association rules has the ability to discover the correlation between the set of attributes and class labels. Intrusion detection systems mainly aim at classifying each new connection record into normal or intrusions. So a class association rule mining method is used in the proposed intrusion detection systems.

Evolution algorithms such as GA and GP have been applied to automatically extract rules. GA evolve the association rules during generations. Each rule is encoded as an individual. Thus, GA only evolves a small number of useful rules which is not enough to build the model. GP is an evolution algorithm with tree structure which has high interpretability. But the tree structure brings some problems which reduce the efficiency of the algorithm. As an extend algorithm of GA and GP, Genetic Network Programming(GNP) has the unique directed structure which solves the bloat problem of GP. The characteristic of reusability nodes can extract a large number of rules during generations. Besides, one GNP individual contains one start node, plural processing nodes and judgment nodes. Each attribute in the data set corresponds to a judgment node. The connections of judgment nodes in the directed graph form the antecedent association parts of rules. Processing nodes work as the starting points of extracting rules. Meanwhile, these rules have been labeled classes. In addition, both the node transitions and the node functions of GNP can be evolved during generations, which also contributes to extracting diversified class association rules for intrusion detection.

When a rule mining method is used, it is efficient to deal with binary attributes of data set. As to quantitative attributes, traditional methods simply partitioned them into two or more intervals. But, these methods result in information loss. Even though the utilization of information gain-based sub-attributes can reduce such loss as much as possible, the discretization of the quantitative attributes would lead to underestimate or overestimate the values that are near the borders. Then, a sharp boundary problem comes out. Fuzzy set theory is helpful to solve this problem. Therefore, Fuzzy GNP is

proposed in this thesis. Fuzzy set theory can assist GNP to extract the rule by allowing different degrees of memberships. At the same time, GNP can evolve fuzzy membership functions during generations to extract more rules.

In general, Fuzzy GNP(79) has the following advantages:

- The class association rules are generated automatically with the evolution of GNP without domain knowledge.
- The sharp boundary problem can be also avoided. This problem comes from the discretization of the continuous attributes into intervals, which lead to under or overestimate the values that are near borders.
- The probabilistic node transition of Fuzzy GNP contributes to obtain the diversified rules.
- Each continuous attribute has its own fuzzy membership function. All of the fuzzy membership functions are evolved with GNP evolution.

1.4.3 Pruning Class Association Rules for Intrusion Detection

Different with the frequent item set-based rule mining method which extracts complete set of rules for historic data set, class association rule mining aims to discover an sufficient number of rules for each class. Even though the aims of both rule mining are different, they all have the ability to extract a large number of rules.

Both association rule mining and class association rule mining take support, confidence or χ^2 to extract rules. In order to reduce the number of extracted rules, some methods simply improve the criteria such as large minimum support value. In this case, this small set of rules results in overfitting problem. The smaller minimum values of support, confidence and χ^2 ensure that many useful rules can be extracted. A large number of class association rules leads to lower efficiency. Simultaneously, many redundant and irrelevant information are contained in the large number of rules. Therefore, pruning rules is necessary to improve the efficiency and detection performance of intrusion detection systems.

The proposed rule pruning method in this thesis contains the following advantages.

- In this approach, the analysis of each is not needed. Genetic Algorithm is used to automatically delete the rule which has bad performance.
- Two stages are used. In the first stage, the rule whose average matching degree with data is lower than a threshold, it is regarded as redundant in the first stage. In

the second stage, GA picks up the effective rules among the remaining rules in the first stage.

- The approach also considers the balance of the four evaluation standards.

1.4.4 Building Classification Models for Intrusion Detection

It is important to build efficient classification models for intrusion detection. As the component of an intrusion detection system, a classification model should classify each new connection record into normal, misuse intrusion or anomaly intrusion. Many researchers have tried to utilize various algorithms to build a classification model, which can work in both misuse detection and anomaly detection. However, it is hard to build a unique classification model. The reasons are described as follows. The automatically produced intrusions from Internet may obey a specific distribution. Whereas, it is difficult to find the restricted distribution of intrusions that are manually made. Besides, it is not easy to distinguish normal and anomaly intrusions. Distance is a direct measure to make classification. Distance-based classification method has no unique model. And it has obvious features like it can execute with considering few parameters. To improve the detection performance, it is critical to identify anomaly intrusions from normal and misuse intrusions. Anomaly intrusions are those attacks never seen previously. Only using the patterns of normal cannot identify anomaly intrusions exactly. Making full use of both normal and misuse intrusions can help to improve the detection of anomaly intrusions. Then, the data and its K-closest neighbors could be regarded as a cluster of similar network behaviors. Thus, if the boundary of each cluster of the similar network behaviors is found, it is easy to distinguish a new connection record. In this thesis, two classification methods are proposed to improve the detection performance. The later one is the modification of the distance-based approach.

1.5 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 builds a hybrid framework of GNP-based class association rule mining for intrusion detection utilizing the advantages of both misuse detection and anomaly detection. Chapter 3 introduces how to prune the class association rules using Genetic Algorithms. The simulation results on proposed rule pruning method outperform those without rule pruning. Chapter 4 proposes

the class association rule mining using Fuzzy GNP. And the simulations demonstrate that Fuzzy GNP is efficient to extract sufficient diversified rules for intrusion detection. Chapter 5 proposes a classification algorithm using the distance between the new data and its closest points of classes. Chapter 6 proposes an improved classification approach which uses Gaussian functions to look for the boundary of each cluster of network behaviors and to identify the data as normal, misuse intrusions and anomaly intrusions. The simulations on the benchmark data set demonstrate the improvement of detection comparing with the basic method, Distance-based method and other state of art methods. Finally, chapter 7 summarizes the thesis.

1.6 Conclusions

In this chapter, the problems in computer security were described, which proved that it is urgent to build a reliable intrusion detection system. Then, the taxonomy of intrusion detection was introduced and the advantages and disadvantages of two general intrusion detection techniques were discussed, too. Next, various approaches applied in intrusion detection were presented. When describing the approaches used for intrusion detection systems, their underlying assumptions and their strengths and limitations were also discussed. More importantly, a general introduction was given on the motivations and contributions to this thesis. Finally, how to evaluate the detection ability was explained.

In the following chapters, the proposed methods are introduced in detail and the simulation results are given on benchmark data sets.

2

A Hybrid Framework of Intrusion Detection System using Genetic Network Programming

Generally speaking, misuse intrusion detection is efficient to identify misuse intrusions from the mixture data set of normal and misuse intrusions. But it cannot detect unknown attacks without any pre-collected patterns(80). On the other hand, an anomaly-based system uses different philosophy. It treats any network connection violating the normal profile as an anomaly(81)(82)(83). A network anomaly intrusion is revealed if the incoming connection deviates from the normal profiles significantly. But it is easy to classify normal as anomaly resulting in many false alarms. Therefore, a new intrusion detection system is proposed in this chapter to utilize the positive features of misuse intrusion detection and anomaly intrusion detection.

2.1 Introduction

Various data mining approaches have been proposed and implemented to detect intrusions in computer security. ADAM(Audit Data Analysis and Mining)(84) uses a frequent itemset-based association rule mining algorithm in detection as an online network-based IDS. The framework of ADAM has two phases: training phase and online phase. In the training phase, the attack-free training data is fed to a module whose output is a rule-based profile of normal activities. After that, the produced profile is inputed to another module to perform a dynamic on-line algorithm with association rules. The next-generation intrusion detection expert system(NIDES)(11) performs a

real-time monitoring of user activities on multiple-target systems connected on a network. It consists of a misuse detection component as well as an anomaly detection component. The misuse detection component employs expert rules to define misuse intrusive activities. And anomaly detection component is based on the statistical approach, and it labels activities as attacks if they are largely deviant from the expected behaviors. (24) also proposed an approach to combine the advantages of misuse and anomaly detection. A random forests algorithm is employed in misuse detection to detect misuse intrusions. Then, they used the outlier detection provided by the random forests algorithm to detect anomaly intrusions.

In this chapter, a new intrusion detection system is proposed to combine the advantages of misuse intrusion detection and anomaly intrusion detection. As a rule generator, Genetic Network Programming(GNP) is used to extract enough number of class association rules for intrusion detection. Different from the popular rule mining method Apriori, it is an evolutionary algorithm and automatically extracts the class association rules in evolution.

The features of the GNP-based intrusion detection system compared with other intrusion detection systems are as follows.

1. One of the features is that both normal and misuse intrusion rules are extracted by evolving GNP individuals, which ensures the diversity and quantity of rules.

2. The other is that the mean and standard deviation of the average matching degrees are used to build the classifier which can distinguish normal, misuse intrusion and anomaly intrusion connections simultaneously, which ensures the full usage of the relation between data and two kinds of rules.

The rest of this chapter is organized as follows: the motivations are presented in Section 2.2. Then, KDD Cup 1999 data set used in the experiments is described in Section 2.3. The framework of the intrusion detection system is introduced in 2.4. The simulations are conducted on KDD Cup 1999 data set in Section 2.5. Finally, the conclusions are drawn in Section 2.6.

2.2 Motivations

Conventionally, intrusion detection techniques are divided into misuse intrusion detection and anomaly intrusion detection. Misuse intrusion detection focuses on detecting known intrusion types by matching the patterns of known intrusion types, while anomaly intrusion detection looks for the new observations which deviate from known normal patterns. Therefore, it is easy to misclassify normal and new intrusion types, since new types of intrusions always pretend to be normal behaviors to attack computer systems. And the detection of new intrusion types becomes difficult in misuse intrusion detection, since no patters about new types of intrusions is contained in this kind of system. Combining the two kinds of techniques in a system makes it possible to remedy the defects of two different techniques by making full use of known information.

On the other hand, most of the intrusion detection systems detect intrusions by experts' experiences or rules extracted from historic data. As one of the most popular data mining methods for many applications, association rule mining is used to discover associations or correlations among a set of attributes in data set. In order to discover useful rules from a dense database, genetic algorithm and genetic programming have been applied to association rule mining. GA evolves the rules during generations and individuals or population themselves represent the association relationships. However, it is not easy for GA to extract enough number of useful rules, because a rule is usually represented as an individual of GA. GP improves the interpretability of GA by replacing the gene structures with the tree structures, which enables higher representation ability of association rules. However, due to the characteristic of tree representations, some problems are commonly experienced, such as code bloat, destructive crossover and structural difficulties. As an extended evolutionary algorithm of GA and GP, genetic network programming (GNP) can represent its solutions using directed graph structures. Owing to this feature, GNP can evolve without the bloat problem of GP. In addition, the advantage of GNP applied to rule mining is that it can extract a sufficient number of useful rules for user's purpose rather than to extract all the rules meeting the criteria. Like most of the existing association rule mining algorithms, conventional association rule mining based on GNP is able to extract rules with attributes of binary values. However, in real world applications, database are more likely to be composed of both discrete and continuous values.

2.3 Data Description

The simulations are conducted on the benchmark KDD Cup 1999 intrusion data set(85). Since 1999, this data set has been the most widely used data set for the evaluation of intrusion detection methods. This data set is prepared by Stolfo et al.(86) and is built based on the data captured in 1998 DARPA intrusion detection evaluation program, prepared and managed by the MIT Lincoln Labs. Lincoln Labs set up an environment to acquire raw TCP dump data of nine weeks from a local-area network (LAN) simulating a typical U.S. Air Force LAN. Thus, the 1998 DARPA data set contains about 4 gigabytes of compressed raw(binary) tcpdump data of 7 weeks of the network traffic for training, which can be processed into about 5 million connection records. Similarly, the test data of two weeks is around 2 million connection records.

Stolfo et al. defined higher-level features that help in distinguishing normal connections from attacks for the 1998 DARPA data set and pre-processed it. Therefore, KDD Cup 1999 data set contains about five million connection records as the training data and about two million connection records as the testing data. Each record in the data set represents a connection between two IP addresses, starting and ending at some well defined times with a well defined protocol. Further, with 41 different attributes, every record represents a separate connection. Hence, in my experiments, every record is considered to be independent each other.

The training data is either labeled as normal or as one of the 24 different kinds of attacks. All of the 24 attacks can be grouped into one of the four classes; *Probe*, $Denialof Service$ (DoS), unauthorized access from a remote machine or $RemotetoLocal$ ($R2L$) and unauthorized access to root or $UsertoRoot$ ($U2R$).

Probing is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls.

DoS is a type of attacks in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.

R2L occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine.

U2R is a class of exploit in which the attacker starts out with access to a normal user account on the system (perhaps gained by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system.

Similarly, the testing data is also labeled as either normal or as one of the attacks belonging to the four attack classes. It is important to note that the testing data includes specific attacks which are not included in the training data. This makes the intrusion detection task more realistic(85).

2.4 Class Association Rule Mining and Classification

The proposed hybrid framework generally combines misuse detection and anomaly detection. It can detect misuse intrusions and anomaly intrusions simultaneously. This hybrid intrusion detection system in this chapter contains the two basic components: rule mining and classification. In the rule mining, GNP is used to extract enough class association rules from normal data and misuse intrusion(known intrusion) data. A brand new intrusion should deviate from normal patterns and differentiate from misuse intrusions. Only considering normal patters is not enough to evaluate whether a new connection is anomaly or not. In the classification, the average matching degree is used to project multi-dimensional network connections into a two dimensional space by extracted normal and misuse intrusion rules. Therefore, the combination of misuse detection classifier and anomaly detection classifier is named hybrid classifier/classification.

2.4.1 Sub-Attribute Utilization

Network connections can be represented as a combination of serious features which are mainly divided into discrete ones and continuous ones. The features which are also called attributes usually with two or more possible values, especially for continuous attributes. Therefore, it is important to select an appropriate partition for continuous attributes and ensure the information loss as little as possible simultaneously. In GNP-based class association rule mining, the information gain is utilized to calculate the threshold to maximize the information gain. Then, continuous attribute is split into two sub-attributes depending on whether the value is larger than a threshold or smaller

than the threshold like the example in Fig. 2.1. In Fig. 2.1, by defining a threshold value for continuous attribute "Count", two sub-attributes are obtained. In class association rule mining, the real value of "Count" is compared with the threshold. "larger than" or "equal to" means it should be the first attribute A_1, while "smaller than" means it belongs to the second attribute A_2.

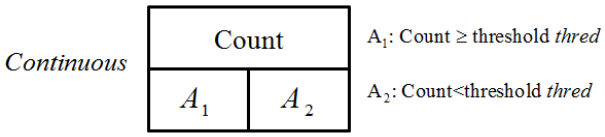


Figure 2.1: Sub-attribute utilization of continuous attributes

In addition, discrete attributes can be divided into binary ones and categorical ones. For binary attribute, it has only two possible values which are 0 and 1. Thus, it has two sub-attributes corresponding to 0 and 1. For categorical attributes, they have two or more possible values such as the attribute which is used to mark the type of network protocal. Such kind of attributes does not consider the relationship since the different values are independent with each other. Therefore, according to the different values, it is partitioned into several sub-attributes. Fig. 2.2 shows the cases of discrete attributes. In Fig. 2.2, the binary attribute *Land* has two possible values of "0" and "1". Therefore, it is divided into two sub-attributes before mining class association rules. Whereas, "*Protocal_type*" is a categorical attribute which has more than two possible values. If it only has three possible values, it will be divided into three sub-attributes.

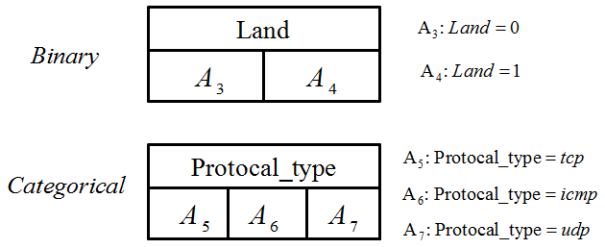


Figure 2.2: Sub-attribute utilization of discrete attributes

2.4.2 Class Association Rule Mining using GNP

An association rule is represented as $X \Rightarrow Y$, where X and Y are the item sets. This rule means that if the items in X exist in a transaction, then Y is also in the transaction

Table 2.1: An example of data set with boolean variables

TID	A_1	A_2	A_3	A_4	C
1	1	0	1	0	0
2	0	1	1	1	1
3	1	1	1	0	1
4	0	1	0	1	1

with high probability. Class association rule is an association rule whose consequent part is restricted to a given class label. GNP-based class association rule mining is first proposed by K. Shimada(87) to do data mining on the data set with boolean variables. Table 2.1 shows an example of data set with boolean variables.

Let A_i be an item in data set and its value be 1 or 0, and C be the class label. TID means the identification number of the tuples in data set. A class association rule can be represented in the following:

$$(A_p = 1) \wedge ... \wedge (A_q = 1) \Rightarrow (C = k), (k = 0, 1, 2, ..., K). \tag{2.1}$$

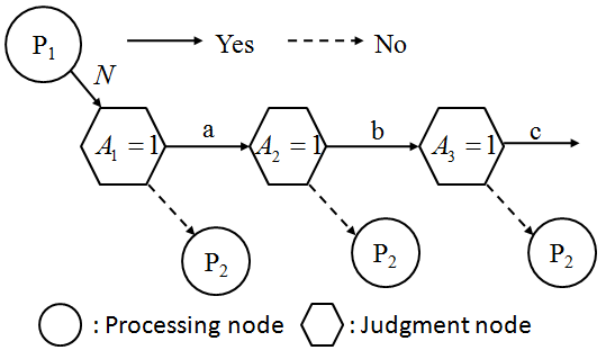


Figure 2.3: GNP-based class association rule mining for boolean variables

GNP examines the attribute values of tuples of data set using judgment nodes and calculates the measurements of association rules using processing nodes. Judgment node determines the next node by a judgment result of Yes or No corresponding Yes-side or No-side as shown in Fig. 2.3. In addition, each judgment node examines the corresponding class label at the same time. Yes-side of the judgment node is connected to another judgment node. Judgment nodes can be reused and shared with other association rules because of GNP's characteristic. No-side of the judgment node is

connected to another processing node, which represents the end of this rule and the start of another new rule. The start node is connected to the first processing node.

The fundamental difference from the evolutionary way of other evolutionary algorithms is that GNP individuals extract interesting rules by evolution and store all the new interesting rules into a pool through the generations(87).

In the extraction of the class association rules for intrusion detection, the attributes and their values correspond to judgement nodes and their judgement values in GNP, respectively. With the sub-attribute utilization, GNP-based class association rule mining successfully combines discrete and continuous values in one single rule. Fig. 2.4 shows an example that GNP extracts candidate class association rules. P_1 is a processing node, which serves as a starting point of class association rule mining and connects to a judgement node. The Yes-side of a judgement node is connected to another judgement node, while the No-side is connected to the next processing node. Judgement nodes here are corresponding to the sub-attributes including both discrete and continuous ones. Taking the above as an example, judgement node A_1 represents the value of the continuous attribute *count* which is greater than or equal to threshold *thred*; A_3 represents the value of the binary attribute *Land* which equals to 0; and A_6 represents the type of the categorical attribute *Protocal* which belongs to *icmp* in the current generation. The total number of tuples moving to Yes-side at each judgement node

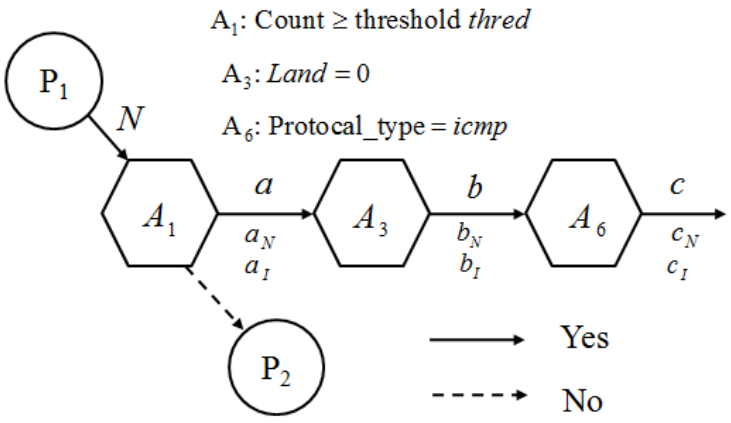


Figure 2.4: GNP representation of class association rules

is calculated at each processing node, which is a start point for class association rule mining.

Table 2.2: *Support* and $confidence$ of class association rules

$Class\ Association\ Rules$	$support$	$confidence$
$A_1 \Rightarrow Normal$	a_N/N	a_N/a
$A_1 \Rightarrow Intrusion$	a_I/N	a_I/a
$A_1 \wedge A_3 \Rightarrow Normal$	b_N/N	b_N/b
$A_1 \wedge A_3 \Rightarrow Intrusion$	b_I/N	b_I/b
$A_1 \wedge A_3 \wedge A_6 \Rightarrow Normal$	c_N/N	c_N/c
$A_1 \wedge A_3 \wedge A_6 \Rightarrow Intrusion$	c_I/N	c_I/c

In Fig. 2.4, N is the number of total tuples. a, b and c are the number of tuples moving to Yes-side at each judgement node. a_N, b_N and c_N are the number of tuples moving to Yes-side at each judgement node which belong to normal class. And a_I, b_I and c_I are those belonging to misuse intrusion class in the application of intrusion detection. Based on the above numbers, the criteria of *support*, $confidence$ and χ^2 can be calculated. Table 2.2 shows how to calculate *support* and $confidence$ values of class association rules. For a rule with the form of $X \Rightarrow Y$, the χ^2 are calculated by $support(X) = x$, $support(Y) = y$ and $support(X \cup Y) = z$ in Eq. 2.2, where, X is the antecedent part of the rule which is represented as a set of attributes and values, and Y is the consequent part of the rule which is represented as class label in class association rule mining.

$$\chi^2 = \frac{N(z - xy)^2}{xy(1-x)(1-y)}. \tag{2.2}$$

The important rules should satisfy the minimum value of *support*, $confidence$ and χ^2. The $support_{min}$, $confidence_{min}$ and χ^2_{min} are given by designers. When an important rule is extracted by GNP, it is checked whether the important rule is new or not. Here, we use $support_{min} = 0.1$, $confidence_{min} = 0.8$ and $\chi^2_{min} = 6.64$ to select important class association rules.

After that, the GNP individuals are evaluated by the fitness defined in the following.

$$F = \sum_{r \in R} \{\chi^2(r) + 10(n_{ante}(r) - 1) + \alpha_{new}(r)\}, \tag{2.3}$$

where R is the set of suffixes of extracted important class association rules in a GNP individual, $\chi^2(r)$ is chi-square of rule r, $n_{ante}(r)$ is the number of attributes of the antecedent of rule r and $\alpha_{new}(r)$ is a constant if rule r is new, otherwise it is 0.

Similar to other evolutionary methods, the structures of GNP are evolved by performing genetic operators. First, the best 1/3 individuals are selected in terms of the

fitness values to do crossover and mutation(87). Two parents selected by tournament selection do crossover and generate two offspring. In detail, each node in parent individuals is selected as a crossover node with the probability of P_c. Then, two parents exchange the genes of the corresponding crossover nodes. Finally, the generated individuals become new ones in the next generation. Whereas, mutation is executed in one individual selected by tournament selection. Two kinds of mutation operators are used in GNP-based class association rule mining. Each branch is selected with the probability of P_{m1} and connected to another node and each node function is selected with the probability of P_{m2} and changed to another one.

The flowchart of class association rule mining using GNP is shown in Fig. 2.5.

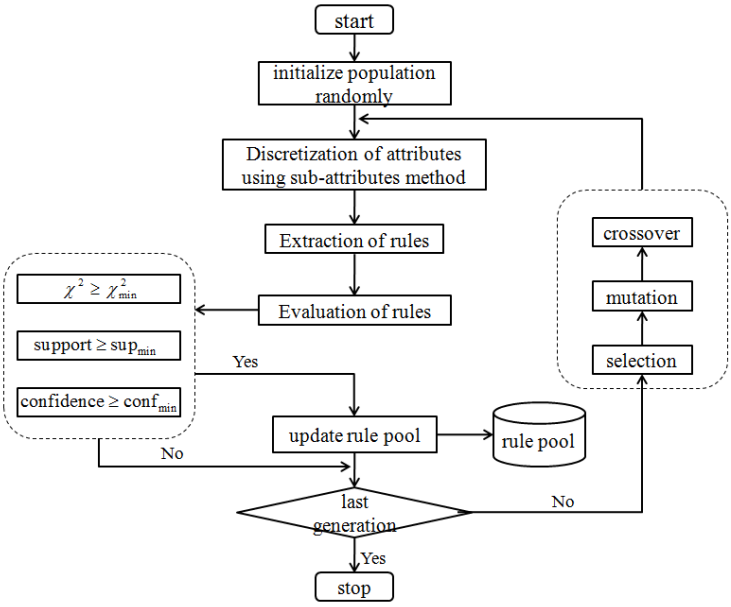


Figure 2.5: Flowchart of GNP-based class association rule mining

2.4.3 Matching Measure using Average Matching Degree

Another distinct difference between the proposed method and other rule-based methods is that a two dimensional space is formed by matching the data with rules. In most of rule-based classification approaches(88)(89)(90)(10), the rules are exploited as descriptive models of different classes. Conventional methods always use a voting scheme to classify a new data according to the classifier which is based on the finite set of

potential rules chosen by a kind of coverage mechanism. The label of the new data corresponds to the best matched rule. While this thesis utilizes a measure to evaluate how well a new data is matched with the rules in each class. Then, a classification model can be built based on the average matching degree space.

Before calculating the average matching degree of training data d with all the rules in each classes, the matching degree of training data d with each rule r in class k is defined as follow:

$$Match_k(d, r) = \frac{N_k(d, r)}{N_k(r)}, \tag{2.4}$$

where, $N_k(d, r)$ is the number of matched attributes of data d with the antecedent part of rule r in class k, $k \in \{normal, misuse\ intrusion\}$. $N_k(r)$ is the number of attributes in the antecedent part of rule r in class k. If $Match_k(d, r)$ equals to 1.0, rule r matches with training data d completely, while, $Match_k(d, r)$ equals to 0, rule r does not match with training data d at all. Then, the average matching degree of training data d with all the rules in class k of the rule pool is defined as follow.

$$m_k(d) = \frac{1}{\mid R_k \mid} \sum_{r \in R_k} Match_k(d, r), \tag{2.5}$$

$m_k(d)$ is the average matching degree of training data d with all the rules in class k. R_k denotes the set of suffixes of rules in class k.

2.4.4 Classification Combining Misuse Detection and Anomaly Detection using Average Matching Degree

Once the multi-dimensional connection data is projected into a two dimensional space, many classification methods can be implemented. In order to compare the detection performances of the proposed hybrid framework and those of the conventional GNP-based misuse detection and GNP-based anomaly detection, the Mean and Standard Deviation Model is used to make the classification.

After obtaining the average matching degree between training data and the rules in class k, we can obtain mean μ and standard deviation σ of the distribution of $m_k(d)$ over all the training data in class k as shown in Eq. 2.6 and Eq. 2.7, respectively.

$$\mu_k = \frac{1}{\mid D_k \mid} \sum_{d \in D_k} m_k(d), \tag{2.6}$$

$$\sigma_k = \sqrt{\frac{1}{|D_k|} \sum_{d \in D_k} (m_k(d) - \mu_k)^2}, \tag{2.7}$$

where, D_k is the set of suffixes of training data in class k. Fig. 2.6 shows one example of the distribution.

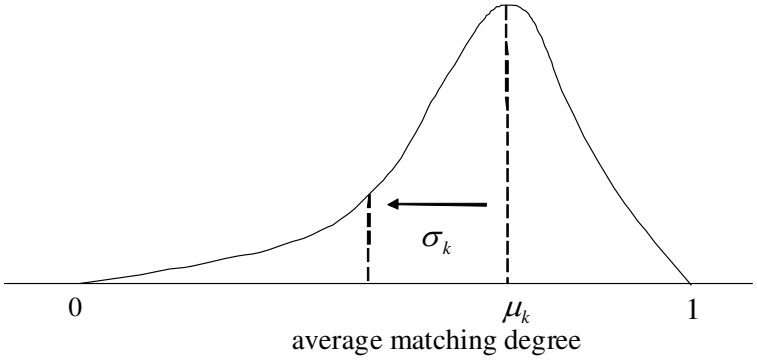


Figure 2.6: An example of mean and standard deviation values of the average matching degree

Now, we have two kinds of means and standard deviations, those are, μ_N and σ_N from normal training data and normal rules, and μ_I and σ_I from misuse training data and misuse intrusion rules, respectively. Fig. 2.7 shows the main idea of the proposed classification method. The horizontal ordinate represents the average matching degree of testing data d_{new} with normal rules, while the vertical ordinate represents the average matching degree of testing data d_{new} with misuse intrusion rules. In the testing period, when a new testing data d_{new} comes, the average matching degree of the new testing data with the rules in the normal rule pool and misuse intrusion rule pool are calculated as $m_N(d_{new})$ and $m_I(d_{new})$. Then,

if $m_N(d_{new}) \leq \mu_N - k_N\sigma_N$ and $m_I(d_{new}) \leq \mu_I - k_I\sigma_I$ then
d_{new} is anomaly intrusion
else
if $m_N(d_{new}) \geq m_I(d_{new})$ then
d_{new} is normal
else
d_{new} is misuse intrusion
end if
end if

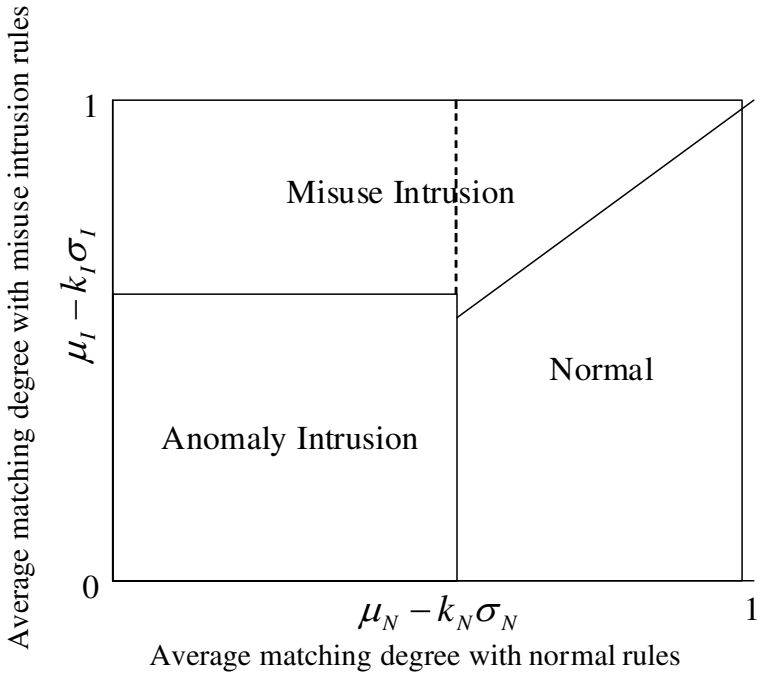


Figure 2.7: Classification combining misuse detection and anomaly detection using average matching degree

Here, k_N and k_I are two parameters that can be adjusted so as to distinguish normal, misuse intrusion and anomaly intrusion for better performance.

The conventional GNP-based misuse detection method classifies the normal connection and misuse intrusion connection by comparing $m_N(d_{new})$ (average matching degree of testing data d_{new} with normal rules) and $m_I(d_{new})$(average matching degree of testing data d_{new} with misuse rules). If $m_N(d_{new}) \geq m_I(d_{new})$, testing data d_{new} is labeled as normal, otherwise, it belongs to misuse intrusions.

However, the conventional GNP-based anomaly detection method carries the GNP-based class association rule mining on normal network behaviors and detects new intrusions by evaluating significant deviations from the normal behaviors. After calculating mean value μ_N and standard deviation σ_N of the distribution of the average matching degree between normal training data with normal rules, if $m_N(d_{new}) \geq \mu_N - k'_N\sigma_N$ is satisfied, testing data d_{new} is labeled as normal, otherwise, d_{new} belongs to anomaly intrusion.

Table 2.3: Parameters of GNP-based rule extraction

Population Size	120
Generation	1000
Processing Node	10
Judgment Node	100
Crossover Rate	1/5
*Mutation Rate*1	1/3
*Mutation Rate*2	1/3

2.5 Simulations

2.5.1 Training Simulations

Using GNP to extract class association rules, the parameters of GNP are set as shown in Table 2.3. In the training step, a sufficient number of data for GNP are prepared to efficiently extract rules. The training data contains 8068 connections randomly selected from KDD Cup 1999 intrusion detection data set, among which 4116 connections are normal and 3952 connections are misuse intrusion including two kinds of attacks, *smurf* and *neptune*, respectively. *smurf* and *neptune* are the misuse intrusion types with the first and second largest numbers in the original database, respectively.

In KDD Cup 1999 database, 41 features are included. After the sub-attribute utilization, 165 sub-attributes are assigned to the judgment functions in GNP. As a result of the evolution, 48,069 rules including 43,342 normal rules and 4,727 misuse intrusion rules are extracted. Fig. 2.8 shows the total number of extracted rules vs. generation.

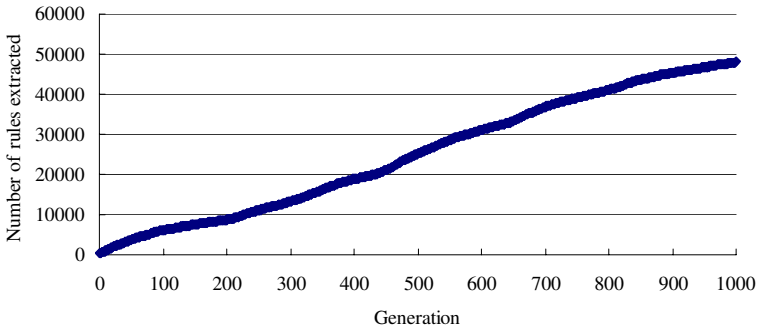


Figure 2.8: Total number of extracted rules by GNP

2.5.2 Testing Simulations

In this simulation, we carry out experiments to verify the effectiveness of the proposed intrusion detection using GNP.

The testing data contains 748 unlabel normal connections and 320 unlabeled intrusion connections. The testing results are given in Table 2.4. In Table 2.4, $Normal(C)$, $Misuse(C)$ and $Anomaly(C)$ indicate the number of normal, misuse intrusions and anomaly intrusions labeled by the basic classifier using both misuse detections and anomaly detections based on the average matching degree, respectively, while $Normal(A)$, $Misuse(A)$ and $Anomaly(A)$ indicate the actual number of normal, misuse intrusions and anomaly intrusions, respectively.

Table 2.4: Classification results of the proposed intrusion detection system

	$Normal(C)$	$Misuse(C)$	$Anomaly(C)$	$Total$
$Normal(A)$	715	11	22	748
$Misuse(A)$	0	188	52	240
$Anomaly(A)$	12	8	60	80
$Total$	727	207	134	1068

From Table 2.4, DR, $Accuracy$, PFR and NFR of the GNP-based intrusion detection system are calculated as follows:

$$\begin{aligned} DR &= (715 + 188 + 60 + 8 + 52)/1068 = 95.79\% \\ PFR &= (11 + 22)/748 = 4.41\% \\ NFR &= (0 + 12)/320 = 3.75\% \\ Accuracy &= (715 + 188 + 60)/1068 = 90.17\% \end{aligned}$$

On the other hand, the proposed GNP-based intrusion detection system is compared with other methods, that is, the conventional GNP-based misuse detection and conventional GNP-based anomaly detection. In the conventional GNP-based misuse detection, only the average matching degrees of a new connection data with normal rules and misuse intrusion rules are compared. It is too simple to classify the new connections. It can get good performances in a data set without anomaly intrusions. Once new types of intrusions are coming, it is difficult to defect. In the conventional GNP-based anomaly detection, it only extracts a pool of normal rules. It has the advantage

that it uses the mean and standard deviation model on the average matching degrees of the training data with normal rules. The mean and standard deviation model requires no priori knowledge about normal activities in order to set thresholds. But only using normal information without any intrusions result in lower DR, higher PFR and NFR.

Fig. 2.9 shows the DR comparison between the proposed GNP-based intrusion detection system and conventional misuse detection and anomaly detection. In misuse detection, the intrusion types in the testing database are the same as those in the training database; while in anomaly detection, the training is an intrusion-free procedure, thus, all the intrusion types can be regarded as anomaly ones. Fig. 2.10 shows the comparisons of PFR and NFR of the proposed intrusion detection system with the conventional misuse detection and anomaly detection.

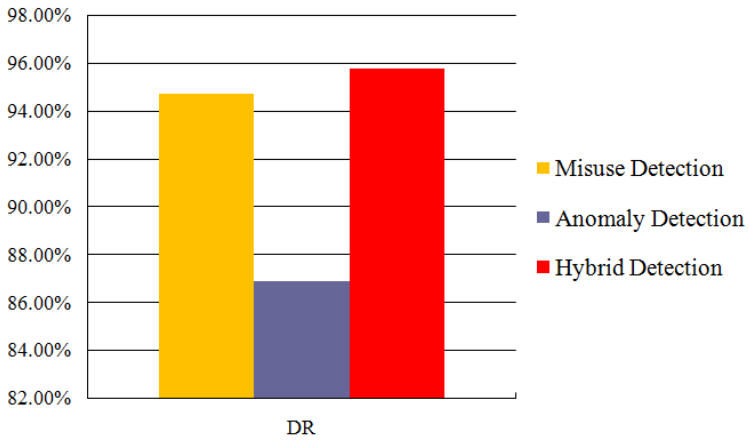


Figure 2.9: Comparisons of DR between the proposed method and conventional methods

From the simulations, it is shown from Fig. 2.9 and Fig. 2.10 that the conventional GNP-based anomaly detection gets much higher PFR. On the other hand, DR is improved by the proposed GNP-based combination detection comparing with conventional GNP-based intrusion detection approaches. The proposed method detects normal, misuse intrusion and anomaly intrusion with higher DR. And the proposed GNP-based intrusion detection system utilizes the advantages of misuse detection and anomaly detection by combing these two methods concurrently. Generally, the proposed system improves the detection performance as well as the balancing between PFR and NFR.

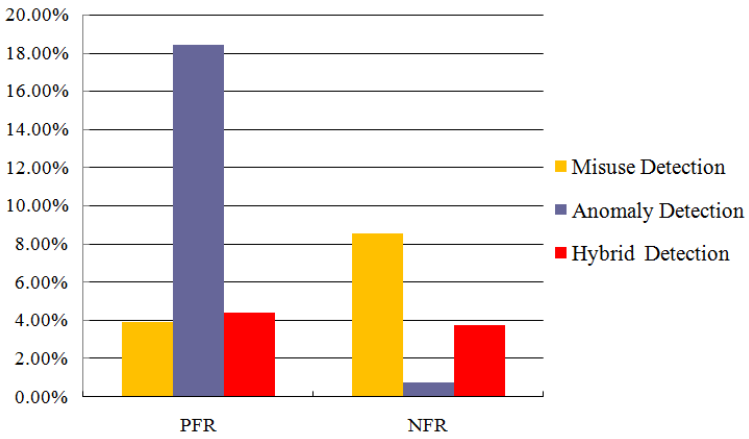


Figure 2.10: Comparisons of PFR and NFR between the proposed method and conventional methods

2.6 Conclusions

In this chapter, a GNP-based intrusion detection system has been proposed to detect misuse intrusions and anomaly intrusions at the same time. The proposed GNP-based intrusion detection system generally consists of class association rule mining and classification. In class association rule mining, information gain is used to partition the continuous features into sub-attributes. Besides, the two dimensional average matching degree space is used to build the classifier for the system. By combing the advantages of misuse detection and anomaly detection, the proposed method has better performances.

However, too many rules bring much useless information into the rule pool inevitably. In this case, an efficient pruning method is needed to reduce the useless rules to improve the efficiency of IDS.

On the other hand, the discretization of the continuous attributes into intervals would lead to underestimate or overestimate the values that are near the borders. This is called the sharp boundary problem.

Then, we solve the two problems separately in the following two chapters.

3

Intrusion Detection System with Rule Pruning using Genetic Network Programming

Class association rule mining is typically known as an important method for data analysis. However, there is a problem in class association rule mining. Even though the goal of the class association rule mining is not to extract a complete set of rules, the number of class association rules extracted is still very large. Therefore, it is time consuming. Most importantly, the rule pool with a large number of rules usually contains much redundant, irrelevant and obvious information. On the other hand, there are two requirements in intrusion detection. One is the real time, the other is detection ability. This leads to two other issues: how can we reduce the number of rules in the model and how can we effectively utilize the rules to make the classification? In this chapter, one of these issues is addressed: the reduction of classification rules. This problem is challenging because the goal is to prune the useless rules while preventing the detection performance of the classifier from dipping.

3.1 Introduction

Most algorithms mining rules for classification aim at finding sufficient rules to cover the training set, rather than finding all the rules from the training data(91). In class association rule mining, the values of the *support*, $confidence$ and χ^2 are main criteria to evaluate whether a rule is important or not. If the minimum values of *support*, $confidence$ and χ^2 are set at high values, the number of the class association rules

extracted is very small. However, the high values of the *support*, $confidence$ and χ^2 result in the overfitting problem to the training data. In addition, some of the important rules with relatively lower values of the *support*, $confidence$ and χ^2 cannot be discovered from training data. In order to extracting useful class association rules, the minimum values of the *support*, $confidence$ and χ^2 should be set at lower values. Then, much redundant and irrelevant information is included in the huge number of rules.

To solve the large number of generated rules, different methods are conducted on the topic of rule pruning. One well known research direction is to remove redundant rules using the concepts(73) of closed item sets and representative rules(74). Some works are proposed to prune the discovered rules in order to form a rule cover(92). Other approaches discard rules that are less relevant with respect to statistical parameters such as the *support*, $confidence$, χ^2 and *conviction*(93). Genetic Relation Algorithm (GRA) as an extension of GP and GNP is used to prune class association rules in (94). It aims at finding different class association rules as many as possible by eliminating similar rules using the fitness calculated by the distances among the rules. The distances among the rules are also calculated by the support values of the rules. However, these approaches lead to information loss. While the manual alteration of the rules needs domain expert knowledge, even if it can lead to significant improvements.

In this chapter, an average matching degree and Genetic Algorithm-based two-stage rule pruning scheme is proposed to remove unnecessary rules as a post-processing procedure. In the first stage, the average matching degree of a data with the rules is calculated and if it is less than a threshold value, then the rule is pruned because it is irrelevant. In the second stage, GA picks up the most interesting rules among the remaining rules in the first stage. Therefore, a reduced set of class association rules is obtained and then, the detection performance is used to evaluate its usefulness.

The main advantages of the proposed method are:

1. It is a post-processing method since its input is a large number of class association rules extracted by GNP.

2. No-prior knowledge from experiences is needed in this two-stage rule pruning method.

3. The first stage is essential in the two-stage rule pruning method because it reduce the the size of GA in the second stage.

4. It is efficient to prune the redundant rules, at the same time, it improves the detection performance of IDS.

5. The proposed method has the ability to control PFR and NFR.

The rest of this chapter is organized as follows. Section 3.2 gives the motivations of this chapter. The two-stage rule pruning method is described in Section 3.3. Simulations are conducted on KDD Cup 1999 in Section 3.4. Finally, we conclude the summary in Section 3.5.

3.2 Motivations

The rule pruning method aims at obtaining the better classification result under a small number of rules. The redundant rules are from two aspects. Firstly, two rules are contradictory. For example, two rules such as $X \Rightarrow C_1$ and $X \Rightarrow C_2$ have the same antecedent part with different class labels, where X represents the antecedent part of the rules and C_1 and C_2 as the consequent parts of the rules mean the class labels of the rules. They are conflict and have no contribution to classification. Secondly, two rules have different generalization levels. Given two rules $R_1 : \ X \Rightarrow C_1$ and $R_2 : \ Y \Rightarrow C_1$ with the same class label and $X \subset Y$, the antecedent part X of R_1 is the subset of the antecedent part Y of R_2 obviously. The general one is kept since longer rules may introduce some redundant information.

In addition, useless rules result in overlapping in two dimensional average matching degree space. For example, an obvious rule contains general knowledge. Thus, it is included in both normal class and intrusion class and always completely matches with the data. In this case, the utilization of such kind of rules will lead to reducing the deviation of the average matching degrees of two entirely different data.

A lot of methods are proposed to prune the redundant and irrelevant rules. Most of the current rule pruning methods are generally divided into two categories. One

category relies on domain knowledge analysis. The category of such methods needs to examine each rule in rule pool one by one. By deleting or reconstructing, then, the number of rules is reduced. The other category relies on sorting the rules in terms of some standards like the *support* and $confidence$. But if two rules have the same antecedent part and identical *support* and $confidence$ with different class labels, they are contradictory rules and cannot be pruned by this category. In addition, all of these methods cannot automatically prune rules. Most importantly, they cannot consider the influence of the rules to the classification. Therefore, an automatic and efficient rule pruning method is motivated by the requirement of reducing the influence of redundant and irrelevant rules in the rule pool.

3.3 Two-Stage Rule Pruning

The average matching degree and Genetic Algorithm-based class association rule pruning is carried out in two stages. Stage I aims to reduce the gene size of GA of stage II, while the objective of stage II is to select a small number of effective rules which improve DR, $Accuracy$, PFR and NFR of the classifier using GA.

3.3.1 Stage I: Average Matching Degree based Method

Roughly speaking, the average matching degree of rule r in class k with the validation data in class k is calculated and if it is less than a threshold value, then the rule is pruned since it is irrelevant.

The average matching degree of rule r in class k with data in class k can be calculated by Eq.(3.1).

$$m_k(r) = \frac{1}{|D_k|} \sum_{d \in D_k} Match_k(d, r), \tag{3.1}$$

$$Match_k(d, r) = \frac{N_k(d, r)}{N_k(r)}, \tag{3.2}$$

where $Match_k(d, r)$ is the matching degree of the rule with each validation data in class k, $N_k(d, r)$ is the number of matched attributes with data d in the antecedent part of rule r in class k and $N_k(r)$ means the number of attributes in the antecedent part of rule r in class k. The mean value μ_k and standard deviation σ_k of $m_k(r)$ over the rules

in class k is calculated as follows:

$$\mu_k = \frac{1}{|R_k|} \sum_{r \in R_k} m_k(r), \tag{3.3}$$

$$\sigma_k = \sqrt{\frac{1}{R_k} \sum_{r \in R_k} (m_k(r) - \mu_k)^2}, \tag{3.4}$$

Therefore, if the average matching degree $m_k(r)$ of rule r with validation data in class k falls in the region less than $\mu_k + k_p * \sigma_k$, that is, $m_k(r) < \mu_k + k_p * \sigma_k$, then rule r is pruned, because it is regarded as redundant.

3.3.2 Stage II: Genetic Algorithm-based Method

In this stage, the binary-coded GA is introduced for further pruning the remaining rules in stage I by encoding the rules as its gene. Gene $v_i(r)$ represents if rule r is pruned or not in the following:

$$v_i(r) = \begin{cases} 1, & if\ rule\ r\ is\ reserved, \\ 0, & otherwise. \end{cases} \tag{3.5}$$

0 represents that the rule is deleted during rule pruning in stage II, while, 1 represents that the rule is used in the classification. Each individual in the population represents a candidate solution for the problem. When evaluating the intrusion detection system, since DR means the ratio that each connection is classified correctly to normal, misuse or anomaly intrusion. ACC means the *Accuracy* which evaluates the ability to classify the normal connections as normal and the intrusion connections as misuse intrusion or anomaly intrusion. PFR means the rate of normal data which are not classified into a normal class and NFR means the rate of intrusion data which are not classified into an intrusion class, PFR increases when more normal connection data are labeled as intrusion, while NFR increases when more intrusion connection data are labeled as normal.

In order to balance these four criteria and get the acceptable DR, *Accuracy*, PFR and NFR, the fitness function of GA is defined as follows.

$$F = \alpha_{DR} * DR + \alpha_{ACC} * ACC - \alpha_{PFR} * PFR - \alpha_{NFR} * NFR \tag{3.6}$$

α_{DR}, α_{ACC}, α_{PFR} and α_{NFR} are the coefficients of DR, Accuracy, PFR and NFR, which can be scaled to guide GA towards required designs.

The initial population of candidate solutions is created randomly. Then, the genetic operators are applied to the individuals in each generation to generate the new population for the next generation.

A. Selection : The selection operation is basically based on the fitness value of each individual. Elite individual is reserved for the next generation, and tournament selection is also used.

B. Crossover : The crossover operator exchanges the gene information of the two parents to create two offspring with the probability of p_c. The two parents are selected by tournament selection among four randomly selected individuals. In this GA, we use the uniform crossover as shown in Fig. 3.1.

Each gene in the offspring is created by copying the gene from one or another parent chosen by a randomly generated binary crossover mask of the same length as the gene. When there is 1 in the crossover mask, the gene is copied from the first parent, and when there is 0 in the mask, the gene is copied from the second parent.

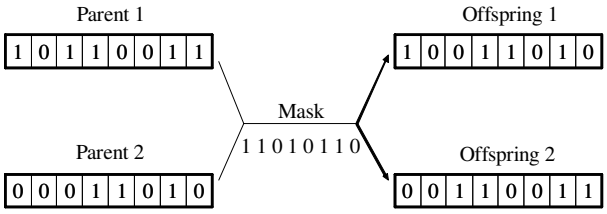


Figure 3.1: Uniform crossover

C. Mutation : The better individual selected by tournament selection is also mutated by mutation operator. Non-uniform mutation operator is used aiming at reducing the disadvantage of random mutation(95). The non-uniform mutation is carried out with mutation rate of p_m, where the operator is defined as follows.

If $s_v^t =< v_1, ..., v_i, ..., v_m >$ is a gene at generation t and element v_i is selected for the mutation, then the following $s_v^{t+1} =< v_1, ..., v_i', ..., v_m >$ is produced,

$$v_i' = mutate(v_i, \bigtriangledown(t, m)), \tag{3.7}$$

$$\bigtriangledown(t, m) = \begin{cases} \lfloor \bigtriangleup(t, m) \rfloor, & if\ a\ random\ digit\ is\ 0, \\ \lceil \bigtriangleup(t, m) \rceil, & if\ a\ random\ digit\ is\ 1, \end{cases} \tag{3.8}$$

where, $mutate(v_i, pos)$ means: mutate the value of the i-th element using the value of position pos of the gene, m is the length of the gene. $\bigtriangledown(t, m)$ is calculated by

Table 3.1: Parameters of GNP-based rule extraction

Population Size	120
Generation	1000
Processing Node	10
Judgment Node	100
Crossover Rate	1/5
*Mutation Rate*1	1/3
*Mutation Rate*2	1/3
χ^2_{min}	6.64
$Support_{min}$	0.1
$Confidence_{min}$	0.8

Eq.(3.8), $\lfloor \triangle(t,m) \rfloor$ means the largest integer not greater than the value of $\triangle(t,m)$, while $\lceil \triangle(t,m) \rceil$ is the smallest integer not less than the value of $\triangle(t,m)$. The $\triangle(t,m)$ is defined as $\triangle(t,m) = m \cdot (1 - r^{(1-\frac{t}{T})^b})$, where r is a random number in (0,1), T is the maximal generation number, and b is a system parameter determining the degree of dependency on the generation number. The function $\triangle(t,m)$ returns a value in the range of $(0,m)$.

3.4 Simulations

The two-stage rule pruning method is evaluated by carrying out the simulations using KDD Cup 1999 data set. Table 3.1 shows the parameters of GNP. And all of the simulations in this chapter are conducted on the same GNP parameters.

3.4.1 Training Simulations

In the training step, a sufficient number of data are prepared for GNP to efficiently extract rules. The training data contains $8,068$ connections randomly selected from KDD Cup 1999 data set, among which 4116 connections are normal and $3,952$ connections are misuse intrusion including two kinds of attacks, *smurf* and *neptune*, respectively. *smurf* and *neptune* are the misuse intrusion types with the first and second largest numbers in the original database, respectively.

In KDD Cup 1999 data set, 41 attributes are included. After the sub-attribute utilization, 165 sub-attributes are assigned to the judgment functions in GNP. As a result of the evolution, $48,069$ rules including $43,342$ normal rules and $4,727$ misuse

intrusion rules are extracted. Thus, many redundant or unimportant rules are also extracted.

3.4.2 Analysis of Two-Stage Rule Pruning Method

This section aims to analyze the two stages of the proposed rule pruning method and check their efficiency on validation data, which contains 4,086 normal connection data and 4,000 misuse intrusion connection data containing 2,000 connection data of *neptune* intrusion type and 2,000 connection data of $smurf$ intrusion type.

Using the training data, the GNP-based class association rule mining method is applied to obtain a large number of class association rules including normal rules and misuse intrusion rules. More rules can ensure more important and interesting information included. However, too many rules also contain much irrelevant and redundant information which can mislead the classification. Then, the average matching degree and GA-based two-stage rule pruning scheme are implemented. Since the gene of GA is encoded by the rules in the rule pool, too many rules mean too large size of genes. Such a huge number of rules becomes an obstacle to realize GA. Therefore, the average matching degree-based rule pruning is implemented in stage I of the two-stage method to reduce the burden of GA-based stage.

Fig. 3.2 shows the number of pruned rules in stage I when using different values of k_p. It is found from Fig. 3.2 that when the value of k_p increases, the number of pruned rules of both normal rules and misuse intrusion rules increases, because the pruning criterion in stage I is strengthened when k_p increases. When the number of the pruned rules is large, the number of reserved rules becomes small in stage I and the structure of GA in stage II becomes simple, which means the complexity of GA will be reduced. Only using stage II will make GA hard to deal with the large rule pool generated by GNP.

In stage II, GA is used to select a reduced number of the best class association rules from the remaining rule pool after executing stage I. In the proposed GA-based rule pruning stage, the population size of GA is 120. The generation number is 100. In genetic operators, $p_c = 0.2$(crossover probability) and $p_m = 0.1$(mutation probability) are used. And Fig. 3.3 shows an example of the fitness curve when k_p equals to 0.9.

On the other hand, α_{DR}, α_{ACC}, α_{PFR} and α_{NFR} of the fitness function can influence the experiments. Fig. 3.4 to Fig. 3.7 show DR, $Accuracy$, PFR and NFR by changing

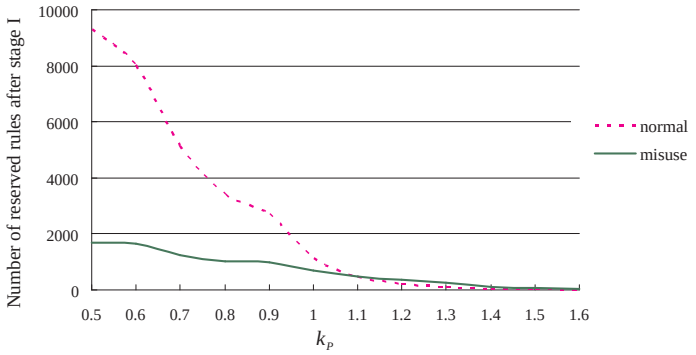


Figure 3.2: Number of reserved rules when k_p varies

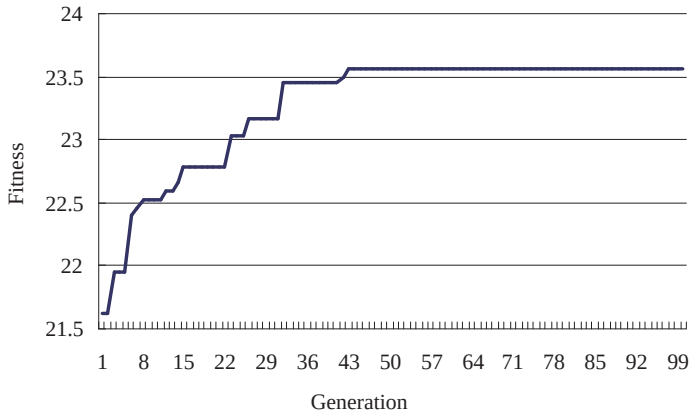


Figure 3.3: The fitness curve of GA in stage II when $k_p = 0.9$

α_{DR}, α_{ACC}, α_{PFR} and α_{NFR} one by one in order to get a set of reasonable coefficients. It is found from Fig. 3.4 that DR and *Accuracy* increase as α_{DR} increases, but PFR decreases, while NFR has no obvious change. Fig. 3.5 shows that *Accuracy* and PFR increase as α_{ACC} increases, but DR is a little bit decreased, while NFR has no change. It is also found from Fig. 3.6 that DR and *Accuracy* decrease as α_{PFR} increases, but PFR decreases, while NFR increases. Fig. 3.7 also shows that DR, *Accuracy* and NFR decrease as α_{NFR} increases, while PFR increases.

Generally speaking, the coefficient of *Accuracy* can improve the accuracy, but it makes DR reduce and PFR increase. Whereas, too much increase of PFR brings the negative influence on the final results, and the improvement of NFR is not obvious

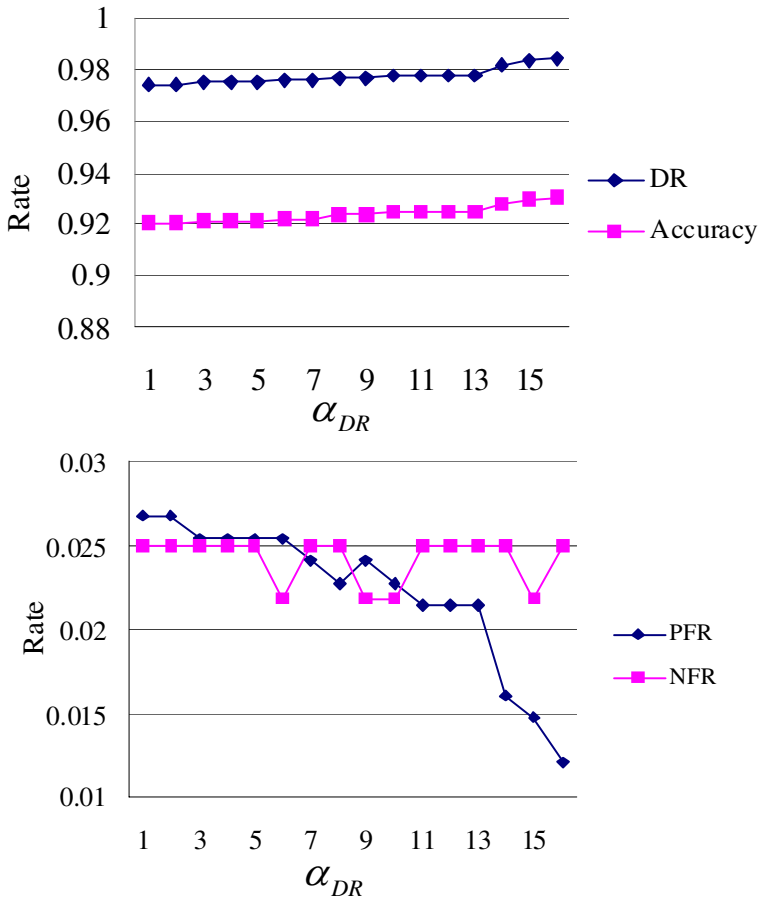


Figure 3.4: DR, *Accuracy*, PFR and NFR versus α_{DR} under $\alpha_{ACC} = 1.0$, $\alpha_{PFR} = 10$ and $\alpha_{NFR} = 10$

when putting too much emphasis on the coefficient of NFR. Therefore, on one hand, it needs to avoid the negative influence on the results brought by the change of coefficients; on the other hand, it needs to balance the relation among DR, *Accuracy*, PFR and NFR in order that the value of each criterion falls in the acceptable range. Here, we used $\alpha_{DR} = 15$, $\alpha_{ACC} = 11$, $\alpha_{PFR} = 30$ and $\alpha_{NFR} = 30$ in the experiments.

Based on the determined coefficients of the fitness function, the effectiveness of the stage I only method, stage I plus stage II method and without pruning are compared from detection performance as shown in Fig. 3.8 and Fig. 3.9. Fig. 3.8 shows the comparisons of DR and *Accuracy* among stage I, stage I plus stage II and without rule pruning, while Fig. 3.9 shows the comparisons of NFR and PFR among stage I, stage I plus stage II and without rule pruning. When the value of k_p is less than 0.9, the

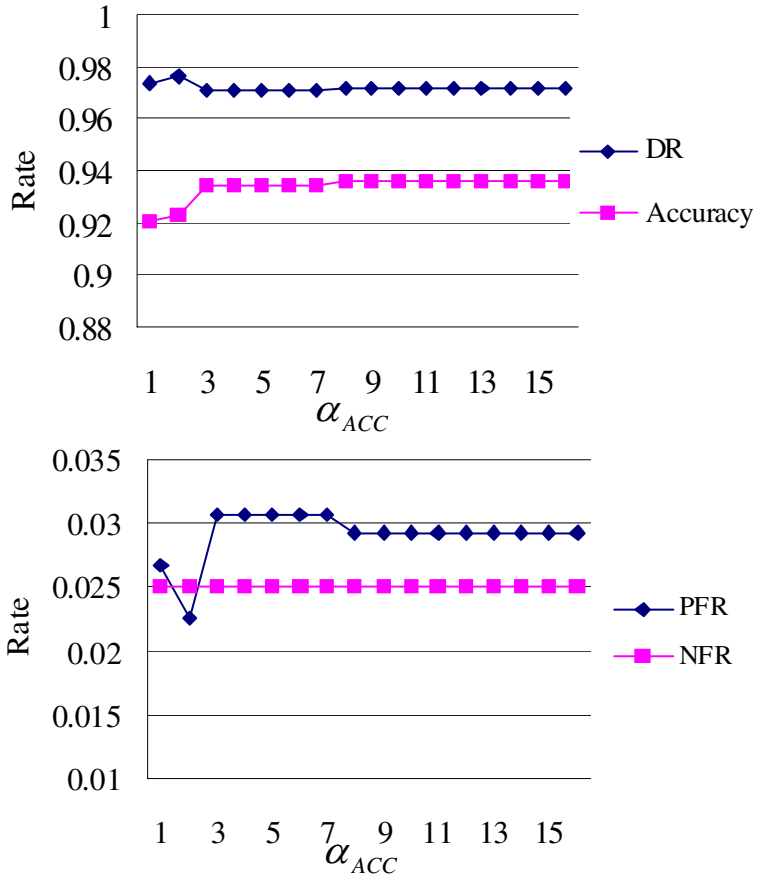


Figure 3.5: DR, *Accuracy*, PFR and NFR versus α_{ACC} under $\alpha_{DR} = 1.0$, $\alpha_{PFR} = 10$ and $\alpha_{NFR} = 10$

results of both DR and *Accuracy* are basically stable. If k_p is larger than 0.9, DR and *Accuracy* sharply decrease, while PFR and NFR also sharply increase, because if the value of k_p is large, too many rules are pruned. The set of rules needs to satisfy not only relevance or importance, but also sufficiency. Stage I makes GA work well to find the best set of rules. However, more strength on stage I tends to prune useful rules, while less strength on stage I makes obviously redundant rules still exist in the rule pool, which brings more burden on GA. More importantly, GA has an important influence on the detection performance. As the results in Fig. 3.8 and Fig. 3.9 show, stage I plus stage II outperforms the performance of only stage I used. The matching degree and GA-based pruning method aims to find the optimal set of rules which is not only interesting or important but also more helpful to direct the classification correctly. It is also proved from Fig. 3.8 and Fig. 3.9 that around $k_p = 0.9$ shows the best performance

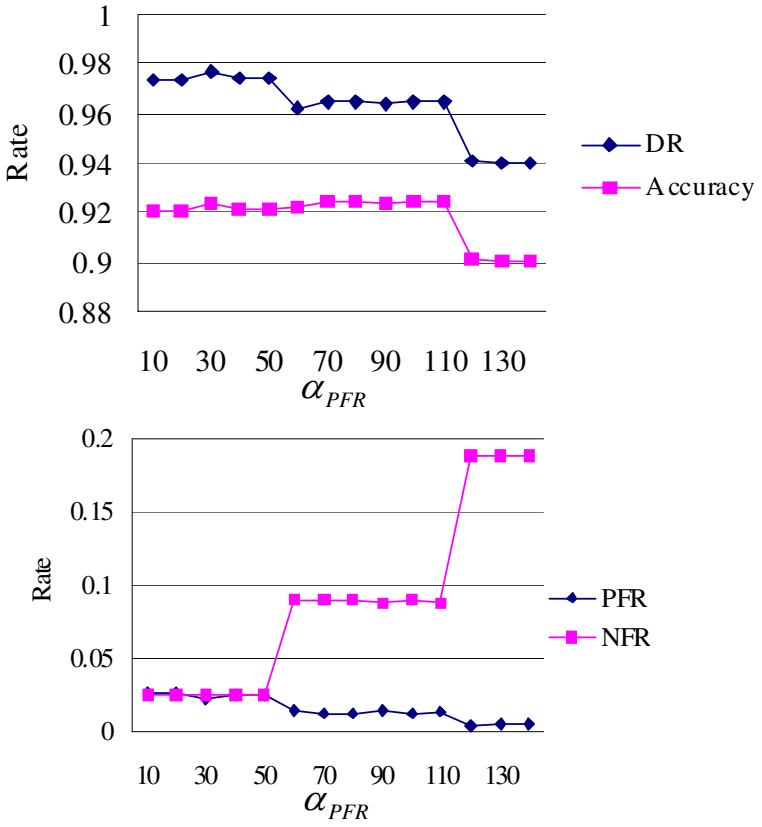


Figure 3.6: *DR*, *Accuracy*, PFR and NFR versus α_{PFR} under $\alpha_{DR} = 1.0$, $\alpha_{ACC} = 1.0$ and $\alpha_{NFR} = 10$

in terms of *DR*, *Accuracy*, PFR and NFR. In addition, the efficiency of two-stage rule pruning is evaluated from time consumption by checking how much time is needed to make classification. As shown in Fig. 3.10, the proposed two-stage rule pruning saves much time for classification, comparing with classification without pruning rules.

3.4.3 Comparisons with Other Methods

In this simulation, the proposed method are compared with conventional GNP-based misuse detection and anomaly detection as well as some other machine learning methods. In the proposed method, we set k_p at 0.9, and 3671 rules are reserved after the first stage rule pruning, which contain 2697 normal rules and 974 misuse intrusion rules. Testing data of the proposed method is the same as that in simulation I. Table 3.2 shows the performance comparison among GNP-based intrusion detection system (GNP-based IDS) with Two-stage rule pruning and different conventional meth-

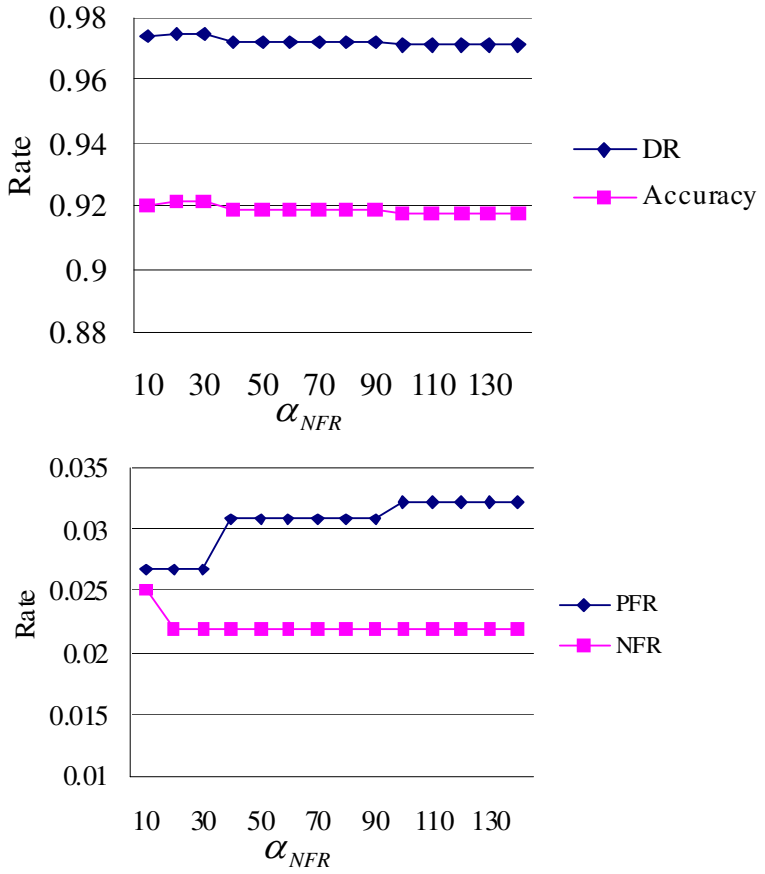


Figure 3.7: DR, *Accuracy*, PFR and NFR versus α_{PFR} under $\alpha_{DR} = 1.0$, $\alpha_{ACC} = 1.0$ and $\alpha_{NFR} = 10$

ods which are GNP-based misuse detection, GNP-based anomaly detection, GP(96), Decision Tree(51) and SVM(97). The first column shows four criteria usually used to evaluate the performance of the intrusion detection systems.

The results show that the proposed method obtains comparative and better performances than conventional GNP-based methods and other classical methods, which means that unified detection is effective and moreover, two-stage rule pruning is also an effective and efficient method.

3.5 Conclusions

In this chapter, a two-stage rule pruning method is proposed to select a small set of useful rules and has improved the detection ability. In the first stage, the average

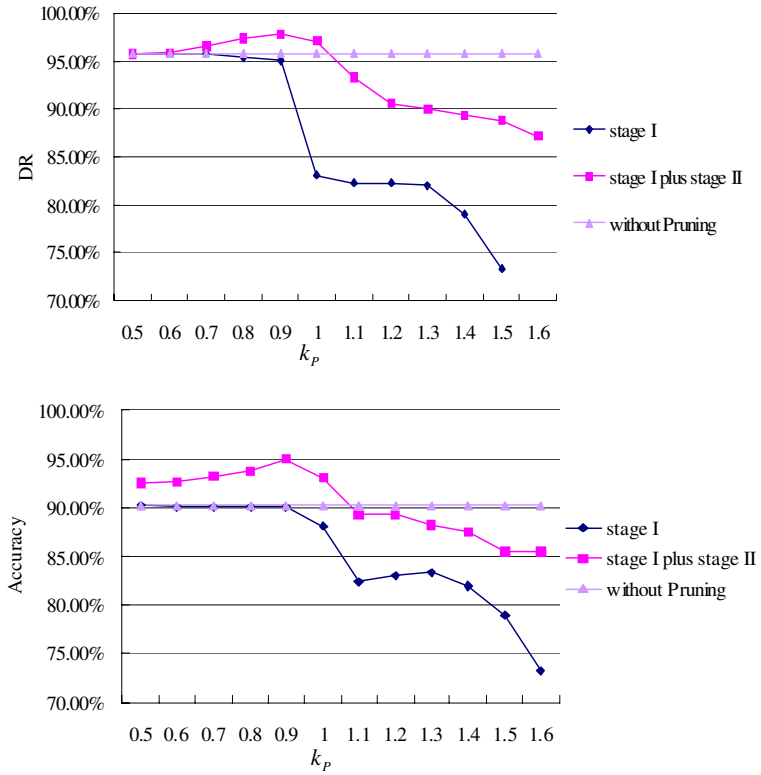


Figure 3.8: Comparison of DR and $Accuracy$ among stage I, stage I plus stage II and without rule pruning

matching degree-based method is used to pre-prune the rules in order to improve the efficiency of GA. In the second stage, GA is implemented to pick up the effective rules among remaining rules in the first stage. In this method, the rules which contribute to high detection performance are selected by GA. The two stages are evaluated using KDD Cup 1999 data set, respectively. The influence of parameters of the two stages are also analyzed. Finally, the performance of the proposed rule pruning method is compared with the performances of other methods. The simulations show that the two-stage rule pruning has better performance from detection ability and time consumption.

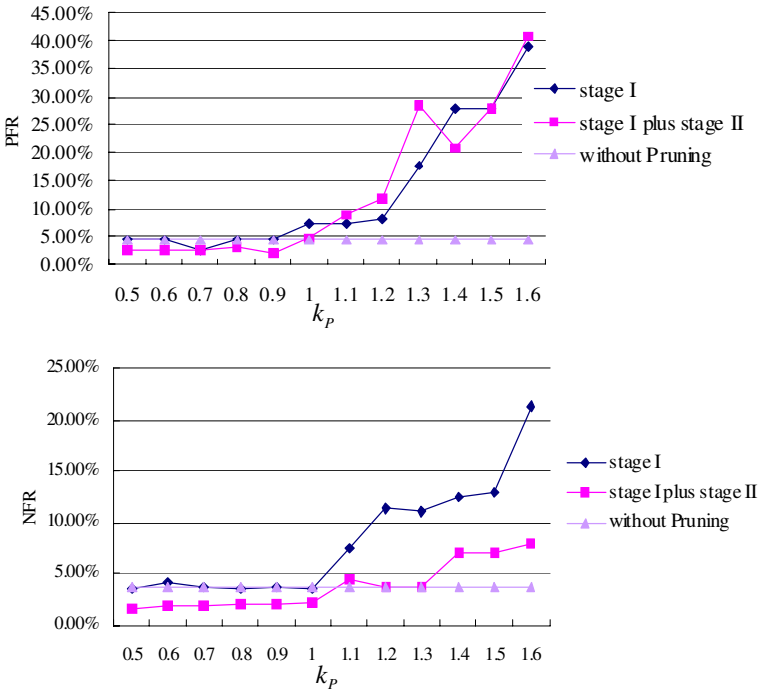


Figure 3.9: Comparison of PFR and NFR among stage I, stage I plus stage II and without rule pruning

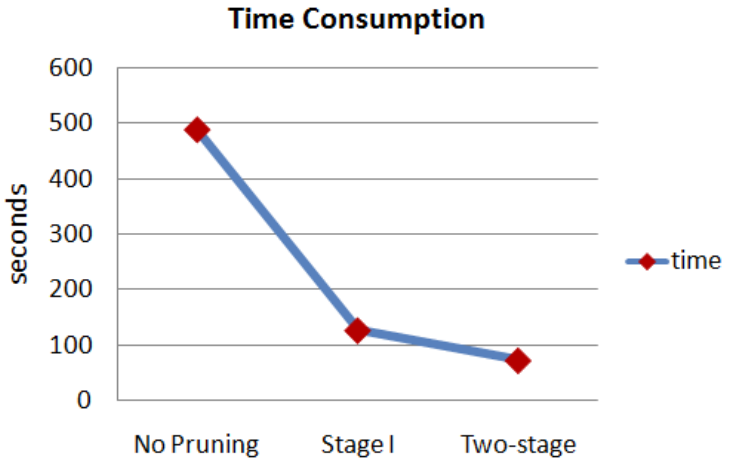


Figure 3.10: Comparison of time consumption among no rule pruning, stage I and two-stage

Table 3.2: The comparisons of the proposed method with other methods

Method	*DR*	*Accuracy*	*PFR*	*NFR*
$GNP-based\ IDS(with\ Two-stage\ Rule\ Pruning)$	97.75%	94.91%	2.01%	2.05%
$GNP-based\ IDS(without\ Two-stage\ Rule\ Pruning)$	95.79%	90.17%	4.41%	3.75%
$GNP-based\ Anomaly\ Detection$	86.89%	–	18.4%	0.75%
$GNP-based\ Misuse\ Detection$	94.71%	–	3.95%	8.54%
GP	90.83%	–	0.68%	–
$Decision\ Trees$	–	89.70%	–	–
SVM	95.5%	–	1.0%	–

4

Intrusion Detection System using Fuzzy Genetic Network Programming

GNP-based class association rule mining initially deals with the data with discrete attributes. For the data with quantitative attributes, it usually divide them into different ranges by crisp discretization. However, for the values near the border, crisp discretization always results in under- or over-estimation in mining process, which is called sharp boundary problem. Fuzzy sets theory introduced by Zadeh(98) in 1965 can resolve this problem by smoothly transferring between member and non-member in fuzzy membership degrees(values). And fuzzy set theory can also bring solution for imprecision and uncertainty information, because of its simplicity and similarity to human reasoning.

4.1 Introduction

The centric topic of data mining is to extract patterns from transaction data in the form of association rules or class association rules. The previous researches on the rule mining usually deal with the database with those attributes having binary or categorical values. But, for the attributes having quantitative values, it is possible to partition the quantitative values to two or more ranges. However, simple partition leads to losing some important information. In order to reducing the information loss as much as possible, the information gain-based sub-attribute method is utilized to deal with continuous attributes. However, the crisp discretization measure to process continuous attributes results in sharp boundary problem, where the discretization of

continuous attributes into intervals would lead to ignore or overemphasize the values that are near boundaries. Fuzzy set theory can help us to overcome this problem by allowing different degrees of memberships. Compared with traditional association rules with crisp sets, the class association rule mining using Fuzzy GNP can provide good linguistic explanation.

In this chapter, the concept of Fuzzy GNP-based class association rule mining is introduced in detail. Fuzzy GNP extracts sufficient class association rules for building the classifiers. It is crucial to extract interesting and sufficient number of class association rules for building the classifiers in intrusion detection, hence, the efficiency of Fuzzy GNP is examined in this chapter. The features of Fuzzy GNP-based class association rule mining are summarized as follows.

1. Experienced and expert knowledge on intrusion detection is not required before the training.

2. Fuzzy GNP can deal with both discrete and continuous attributes in intrusion detection to overtake the sharp boundary problem in sub-attribute method.

3. Fuzzy GNP can extract diversified class association rules by evolving Fuzzy GNP.

4. Probabilistic node transition takes place of the traditional node transition in GNP. This change also contributes to extracting diversified rules.

5. Each continuous attribute has its own initial fuzzy membership function which is different from each other. In addition, the fuzzy membership functions are evolved along with GNP.

The rest of this chapter is organized as follows: the motivations are presented in Section 4.2. In Section 4.3, it is described how Fuzzy GNP is used to extract class association rules. In Section 4.4, the hybrid classification method combing misuse detection and anomaly detection is used on class association rules extracted by Fuzzy

GNP. The simulation results are given on KDD Cup 1999 data set in Section 4.5. Finally, the conclusions are drawn in Section 4.6.

4.2 Motivations

It is easy to deal with discrete attributes in the data when extracting class association rules. But for quantitative attributes, simple discretization leads to the loss of important information. Most importantly, crisp discretization brings sharp boundary problem. Therefore, designing an appropriate discretization for quantitative valued attributes is a challenge in GNP-based class association rule mining. Fuzzy set theory(99)(100) has the ability to solve the sharp boundary problem by representing quantitative values as different fuzzy membership degrees. At the same time, the extracted rules can be represented in a more exact way. And the class association rules become more understandable by introducing fuzzy set theory.

Generally speaking, there are two main reasons to introduce fuzzy set theory into intrusion detection(101). First, many quantitative features are involved in intrusion detection and can potentially be viewed as fuzzy variables. For instance, the CUP usage time and the connection duration are two examples of quantitative features which can be viewed as fuzzy variables. The second reason for introducing fuzzy set theory to intrusion detection is that security itself includes fuzziness. Sometimes, the value of the quantitative attribute can decide the data is more like normal or intrusion. The fuzzy membership degrees can show how much it belongs to normal and how much it belongs to intrusion.

By introducing fuzzy set theory into GNP, probabilistic node transition is used, which improves the exploration ability of useful class association rules. In this way, much more diversified rules can be extracted. Both quality and quantity of extracted rules can be improved by Fuzzy GNP. Hence, it can ensure the detection ability of intrusion detection.

4.3 Class Association Rule Mining using Fuzzy GNP

4.3.1 Fuzzy Membership Functions for Continuous Attributes

Firstly, the values of all continuous attributes of the database are fuzzified into three linguistic terms like Low, Middle and High as shown in Fig. 4.1. These linguistic

terms are defined by the combination of trapezoidal and triangular membership functions symmetrically spaced. Each continuous attribute is associated with its own fuzzy membership function.

Fuzzy set theory came from the desire to describe complex systems with linguistic descriptions. In most of the real-world applications such as data mining, the types of data are complex, i.e., not only the type of boolean, but also many other types are also included. In this chapter, the advantage of fuzzy set theory is used allowing every continuous attribute to have fuzzy membership values in $[0, 1]$. Furthermore, each continuous attribute in the database is transformed into three linguistic terms (Low, Middle and High). Besides, the membership function of each continuous attribute is evolved generation by generation in order to discover many interesting rules.

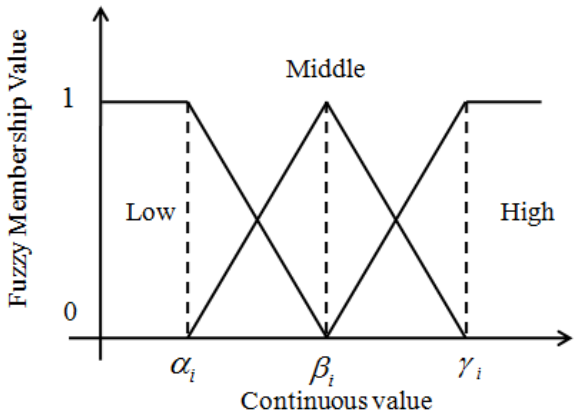


Figure 4.1: Fuzzy Membership Function of attribute A_i

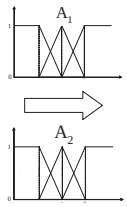
Actually, the parameters of α_i, β_i and γ_i of fuzzy attribute A_i shown in Fig. 4.1 are also evolved along with GNP evolution. Once a GNP individual starts the searching for class association rules, the fuzzy membership values are used to determine the transitions in GNP individuals.

4.3.2 Class Association Rule Mining using Fuzzy GNP

Considering sub-attribute utilization mechanism, binary attribute *Land* is divided into two sub-attributes $Land = 1$ and $Land = 0$. The symbolic attribute is divided into two or more sub-attributes. As to the continuous attributes shown in Fig. 4.2, attribute

A_1 and A_2 can be transformed into fuzzy membership values by the fuzzy membership functions.

TID	A_1	A_2
1	2500	23
2	4800	34
3	3200	36
4	5600	50



TID	A1			A2		
	A_1_Low	A_1_Mid	A_1_High	A_2_Low	A_2_Mid	A_2_High
1	0.75	0.25	0.00	0.85	0.15	0.00
2	0.00	0.60	0.40	0.30	0.70	0.00
3	0.40	0.60	0.00	0.20	0.80	0.00
4	0.00	0.20	0.80	0.00	0.50	0.50

Figure 4.2: An example of transformation of continuous attributes

When a judgment node represents continuous attribute A_i with linguistic term $Q_i \in \{Low, Middle, High\}$, the transition from the current judgment node to the next node is determined by the membership value of the fuzzy attribute A_i with linguistic term Q_i(30).

Class association rule mining based on Fuzzy GNP successfully combines fuzzy set theory with conventional class association rule mining, which utilizes discrete attributes and continuous attributes in one single rule. Fig. 4.3 shows how class association candidate rules extract using Fuzzy GNP. In Fig. 4.3, processing node P_1 serves as the starting point of the class association rule mining and connects to a judgement node. One judgment node has two possible branches, Yes-side branch and No-side branch. As Fig. 4.3 shows, the first judgement node J_1 examines the judgment function whether the discrete attribute *Land* equals to 1. If it equals to 1, J_1 would connect to another judgment node which is J_2 in Fig. 4.3. If it does not equal to 1, then a candidate rule which contains only one item in the antecedent part is extracted. Then, judgment node J_2 checks the function whether the discrete attribute *service* equals to *http*. If the judgment function is not satisfied, one more candidate rule is extracted, which has two items in the antecedent part. Otherwise, the transition is done to another judgment node. In Fig. 4.3, J_3 would check the continuous attribute $count = Low$, then Fuzzy GNP is used to decide which branch should be selected. At the end of evolution, Fuzzy GNP automatically counts the number a, b and c, which are the numbers of tuples moving to Yes-side at the judgment nodes. a_N, b_N, c_N are those with *Normal* class in the application of intrusion detection. a_I, b_I, c_I are those with *Misuse Intrusion*

class. Then, the criteria of the *support*, $confidence$ and χ^2 can be calculated using the values counted by Fuzzy GNP.

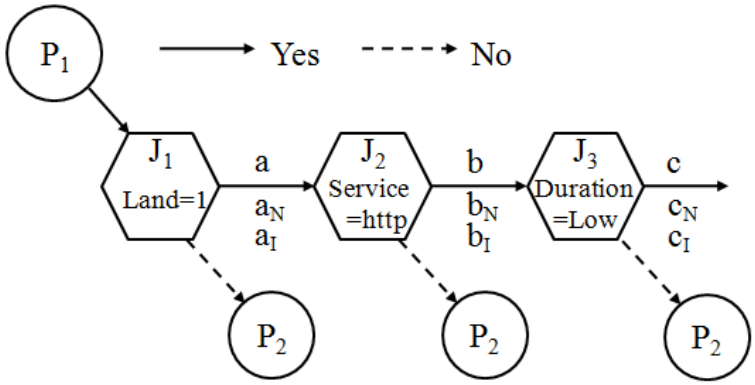


Figure 4.3: Mining class association candidate rules using Fuzzy GNP

There are two main differences between Fuzzy GNP and regular GNP. One is the functions of judgment nodes. This difference means that the functions of judgment nodes examined should be about continuous attributes. The other is the node transition, which means that once continuous attributes are examined, the probabilistic node transition should be used to decide Yes-side or No-side.

4.3.3 Probabilistic Node Transition in Fuzzy GNP

Fig. 4.4 and Fig. 4.5 show how the two kinds of probabilistic transition of judgment nodes are done in Fuzzy GNP individuals, simple probabilistic transition and accurate probabilistic transition, respectively.

In the simple probabilistic transition, if the membership value $\mu_{Q_i}(a_i)$ of continuous attribute A_i with linguistic term Q_i is greater than or equal 0.5, then go to the Yes-side of the judgment node, otherwise, go to the No-side of the judgment node as shown in Fig. 4.4, where a_i is the value of continuous attribute A_i.

In the accurate probabilistic node transition, a random value r is generated uniformly and compared with the fuzzy value of the continuous attribute like Fig. 4.5 shows.

In the accurate probabilistic transition, a random number r is generated and compared with the membership value of the continuous attribute in Fig. 4.1. If the random

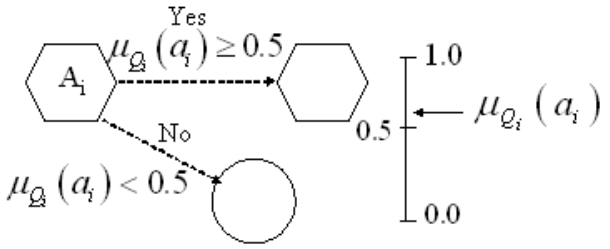


Figure 4.4: Simple probabilistic node transition from one judgment node to another

number is smaller than or equal to the membership value $\mu_{Q_i}(a_i)$, then go to the Yes-side of the judgment node, otherwise, go to the No-side of the judgment node.

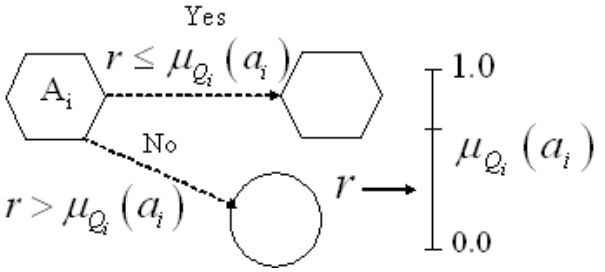


Figure 4.5: Accurate probabilistic node transition from one judgment node to another

4.3.4 Mutation of Fuzzy Membership Function

At the beginning of evolving fuzzy membership function (FMF), the parameters of α_i and β_i for attribute A_i should be initialized by analyzing the distribution of the data, γ_i is automatically calculated by $(2\beta_i - \alpha_i)$.

Then, nonuniform mutation is used to adjust the parameters α_i, β_i for each continuous attribute A_i. During the process of evolving, the parameters are selected by non-uniform mutation with the probability of P_m. Let x_k be a parameter selected for mutation in the kth generation, then we can calculate x_{k+1} as follows.

$$x_{k+1} = \begin{cases} x_k + \Delta(k, UB - x_k), & when\ \epsilon\ is\ 0; \\ x_k - \Delta(k, x_k - LB), & when\ \epsilon\ is\ 1; \end{cases} \tag{4.1}$$

where, UB and LB are the upper and lower bounds of variable x_k, and ϵ is a random binary value in $\{0, 1\}$. The function $\Delta(k, y)$ returns a value in $(0, y)$, where $\Delta(k, y)$ approaches 0 as k increases. Such property causes the operator to search the space

uniformly at first and very locally in the later generations. The actual $\Delta(k, y)$ can be calculated by the following equation:

$$\Delta(k, y) = y(1 - r^{(1-k/T)^{\eta}}), \tag{4.2}$$

where, r is a uniform random number in $(0, 1)$, T is the maximal generation number and η is a system parameter determining the degree of dependency on the generation number.

4.3.5 Fitness Function and Genetic Operators in GNP

In order to selecting the interesting class association rules for classifier, we define the fitness function of GNP individuals by

$$F = \sum_{r \in R} \{\chi^2(r) + 10(n(r) - 1) + \alpha_{new}(r)\}, \tag{4.3}$$

where, R is a set of suffixes of extracted important rules, $\chi^2(r)$ is the χ^2 value of rule r, and $n(r)$ is the number of attributes in the antecedent part of rule r. $\alpha_{new}(r)$ is an additional constant defined by the following equation

$$\alpha_{new}(r) = \begin{cases} \alpha_{new}, & \textit{when rule } r \textit{ is newly extracted}, \\ 0, & \textit{otherwise}. \end{cases} \tag{4.4}$$

$\chi^2(r)$, $n(r)$ and $\alpha_{new}(r)$ are concerned with the importance, complexity and novelty of rule r, respectively.

In each generation, the individuals are ranked by their fitness values and upper 1/3 individuals are selected, then they are reproduced by three kinds of genetic operators for the next generation.

Crossover is executed between two parents and two offspring is generated. In detail, each node in parent individuals is selected as a crossover node with the probability of P_c. Then, two parents exchange the genes of the corresponding crossover nodes. Finally, the generated individuals become new ones in the next generation.

Whereas, mutation is executed in one individual. Two kinds of mutation operators are also used to evolve Fuzzy GNPs. Each branch is selected with the probability of P_{m1} and changed to another node and each node function is selected with the probability of P_{m2} and changed to another one.

The flow chart of Fuzzy GNP class association rule mining is shown in Fig. 4.6.

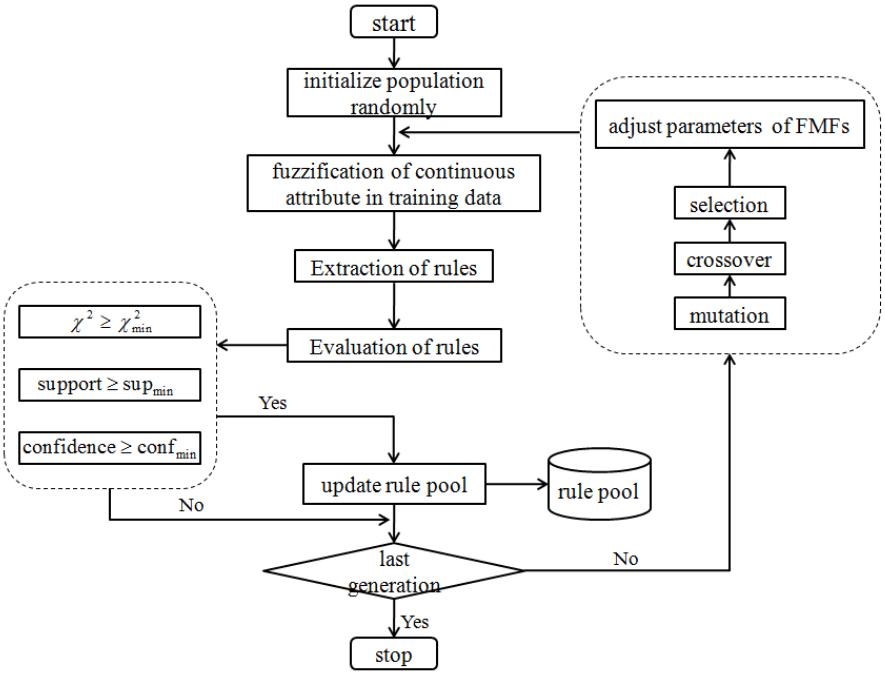


Figure 4.6: Flow chart of class association rule mining using Fuzzy GNP

4.4 Building Classifier

4.4.1 Matching Measure

The first step of the classification is to calculate the average matching degree of each connection data with the rules of each class, which can solve the problem that the conventional matching measure easily favors the minority too much(102). Different from the GNP-based class association rule mining in chapter 2, the matching degree in this chapter can be divided into two parts according to continuous attributes and discrete attributes in the class association rules extracted by Fuzzy GNP. The matching degree for continuous attribute can be calculated using Eq.(4.5).

$$MatchDegree(Q_i, a_i) = \mu_{Q_i}(a_i), \tag{4.5}$$

where, Q_i means the linguistic term of continuous attribute A_i in rule r. a_i means the value of the attribute A_i. μ_{Q_i} represents the membership function for linguistic term Q_i.

Then, the whole matching degree of data d with rule r in class k(including p continuous attributes and q discrete attributes) is defined by:

$$Match_k(d,r) = \frac{1}{p+q}(\sum_{i \in CA} \mu_{Q_i}(a_i) + t), \tag{4.6}$$

where, CA is the set of suffixes of continuous attributes in rule r of class k, and t is the number of matched discrete attributes in rule r of class k with data d. As a result, the average matching degree can be defined by

$$m_k(d) = \frac{1}{|R_k|} \sum_{r \in R_k} Match_k(d,r), \tag{4.7}$$

where, R_k is the set of suffixes of the extracted rules in class k in the rule pool.

4.4.2 Classification based on the Average Matching Degree

The classification method is the same as the one used in chapter 2. After obtaining the average matching degree between training data and the rules in class k, the mean μ and standard deviation σ of the average matching degrees over all the training data in class k as shown in Eq.(4.8) and Eq.(4.9), respectively.

$$\mu_k = \frac{1}{\mid D_k \mid} \sum_{d \in D_k} m_k(d), \tag{4.8}$$

$$\sigma_k = \sqrt{\frac{1}{|D_k|} \sum_{d \in D_k} (m_k(d) - \mu_k)^2}. \tag{4.9}$$

where, D_k is the set of suffixes of training data in class k.

Then, μ_N and σ_N represent the distribution of the average matching degree of normal training data with normal rules, and μ_I and σ_I represent that of misuse training data with misuse intrusion rules. When a new testing data d_{new} comes, the average matching degree of the new testing data with the rules in the normal rule pool and misuse intrusion rule pool are calculated as $m_N(d_{new})$ and $m_I(d_{new})$. According to the procedure explained in chapter 2, new data d_{new} can be distinguished as normal, misuse intrusion and anomaly intrusion.

Table 4.1: Parameters of Fuzzy GNP-based rule extraction

Population Size	120
Generation	1000
Processing Node	10
Judgment Node	100
Crossover Rate	1/5
*Mutation Rate*1	1/3
*Mutation Rate*2	1/3

4.5 Simulations

The simulations are conducted on the intrusion detection database of KDD CUP 1999. In order to analyze the proposed method, Fuzzy GNP based rule mining with simple probabilistic node transition and that with accurate probabilistic node transition will be studied in this section. In addition, the analysis of parameters are done. And the effect of Fuzzy GNP and and that of GNP with the hybrid classifier are compared using the average matching degree space.

4.5.1 Performance of Fuzzy GNP Mining Class Association Rule with Two Kinds of Node Transitions

10,000 network connections are randomly chosen from KDD CUP data set as the training data which consists of 5,000 normal network connections and 5,000 intrusion network connections including two kinds of attacks, *neptune* and *smurf*. The parameters of Fuzzy GNP are summarized in Table 4.1. A new rule, which satisfies the minimum support, confidence and χ^2, is extracted. The minimum confidence is set at 0.8 and the minimum χ^2 is set at 6.63. The minimum support is set at 0.1 and 0.075 for normal and intrusion rules, respectively.

After 1,000 generations, Fuzzy GNP extracts 46,346 and 61,628 rules using simple probabilistic node transition and accurate probabilistic node transition, respectively. The conventional GNP extracts 26,433 rules from the same training data. Fig. 4.7 shows the number of extracted rules versus generation. In Fig. 4.7, the boldest line in blue represents the number of extracted rules by Fuzzy GNP with accurate probabilistic node transition. The bold line in green represents the number of extracted rules by Fuzzy GNP with simple probabilistic node transition. And the normal line in red represents the number of extracted rules by conventional GNP. The numbers of

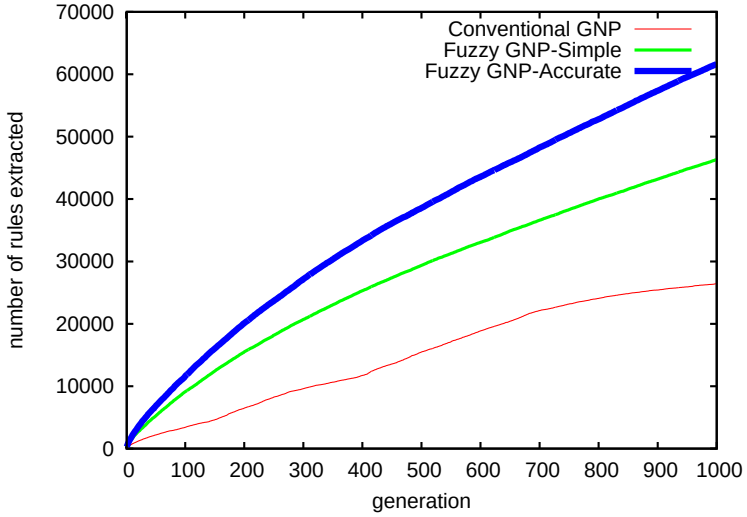


Figure 4.7: Total number of extracted rules by GNP

extracted rules in Fig. 4.7 indicate that Fuzzy GNP has the ability to extract much more rules than conventional GNP. And the accurate probabilistic node transition can contribute to extracting more class association rules than the simple one.

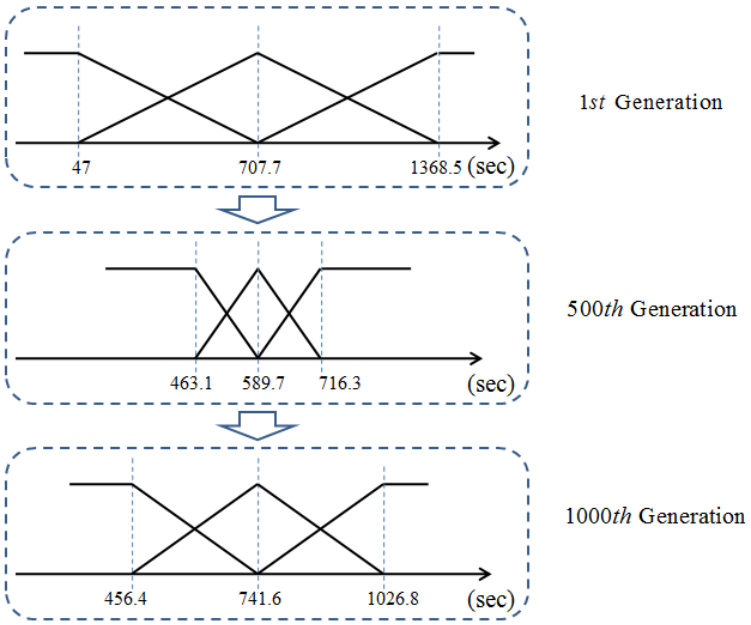


Figure 4.8: The evolution of fuzzy membership function of attribute "*duration*"

Fig. 4.8 shows an example of the evolution of the fuzzy membership function. The

first fuzzy membership function is the initial one. The second is the one evolved after 500 generations. The third is the final fuzzy membership function.

In this case, the settings of the parameters k_I and k_N are very important because the classifier must classify three kinds of data including anomaly intrusion simultaneously. So, first, the analysis of the parameters is given by the simulations using a validation data set, and the best parameters for the testing are determined.

1) Analysis and determination of the weight parameters in the validation: The validation data contains 748 normal connection data, 240 misuse intrusion connection data and 80 anomaly intrusion data which are not contained in the training data. First, the effects of the parameters k_I and k_N are analyzed on the class association rules extracted by Fuzzy GNP with simple probabilistic node transition and that with accurate probabilistic node transition, respectively. Fig. 4.9 shows DR, Accuracy, PFR and NFR based on Fuzzy GNP with simple probabilistic node transition and Table 4.2 shows the corresponding parameter settings use in Fig. 4.9. Setting 1-6 shows that, as

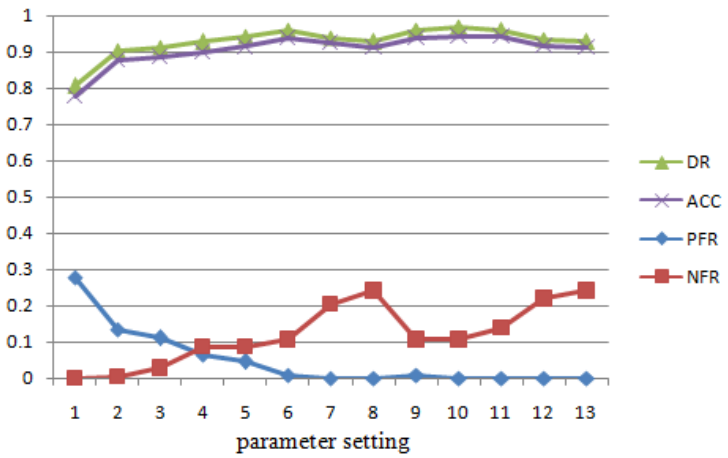


Figure 4.9: Effects of the parameter settings on DR, ACC, PFR and NFR in the case of simple probabilistic node transitions

k_N increases, DR, Accuracy and PFR are improved although NFR is increased. After that, DR, Accuracy are decreased and NFR becomes worse with the increase of k_N. And the improvement of PFR is not obvious. Thus, k_N is set at 3. Then, with the increase of k_I, DR and Accuracy are increased at the setting 9, 10. And NFR is better than that in the setting 7, 8. At the beginning of setting 11, DR and Accuracy become

Table 4.2: Parameter settings in the case of simple probabilistic node transition

setting	k_I	k_N
1	1.0	0.5
2	1.0	1.0
3	1.0	1.5
4	1.0	2.0
5	1.0	2.5
6	1.0	3.0
7	1.0	3.5
8	1.0	4.5
9	1.5	3.0
10	2.0	3.0
11	2.5	3.0
12	3.0	3.0
13	3.5	3.0

lower while NFR is increased and PFR is not changed. Therefore, k_I and k_N in the case of simple probabilistic node transition is set at 2.0 and 3.0, respectively.

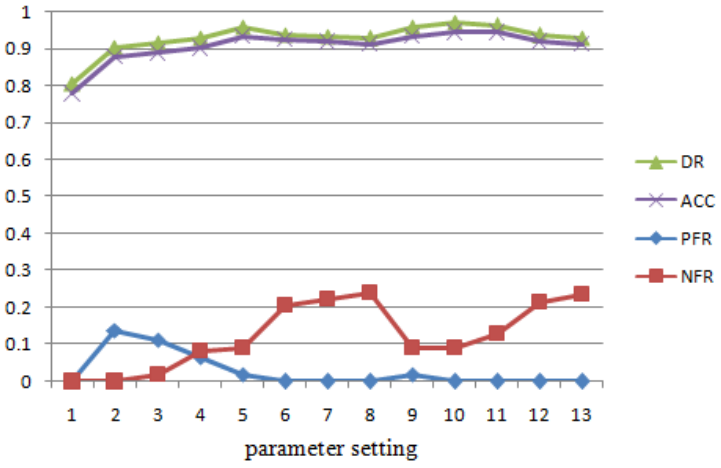


Figure 4.10: Effects of the parameter settings on DR, ACC, PFR and NFR in the case of accurate probabilistic node transitions

Fig. 4.10 shows DR, Accuracy, PFR and NFR based on Fuzzy GNP with simple probabilistic node transition and Table 4.3 shows the corresponding parameter settings used in Fig. 4.10. The settings 1-5 indicate the detection ability except NFR is increased with k_N increases. When k_N is increased more like the settings 6-8, the detection ability becomes worse. So, k_N is fixed at 2.7 as setting 5 and the effect of k_I is examined. At

Table 4.3: Parameter settings in the case of accurate probabilistic node transition

setting	k_I	k_N
1	1.0	0.5
2	1.0	1.0
3	1.0	1.5
4	1.0	2.0
5	1.0	2.7
6	1.0	3.5
7	1.0	4.1
8	1.0	4.6
9	1.5	2.7
10	2.0	2.7
11	2.7	2.7
12	3.1	2.7
13	3.5	2.7

the beginning, the increase of k_I improves the detection ability. Till the setting 10, the detection performance becomes bad in the setting 11-13. By the analysis, k_I and k_N in the case of accurate probabilistic node transition are set at 2.0 and 2.7.

Table 4.4: Classification results of Fuzzy GNP with simple probabilistic node transitions

	$Normal(C)$	$Misuse(C)$	$Anomaly(C)$	$Total$
$Normal(A)$	1928	1	1	1930
$Misuse(A)$	166	7444	278	7888
$Anomaly(A)$	130	7	45	182
$Total$	2224	7452	324	10000

2) Testing results: The testing data set contains 10,000 connection data, where 8 new types of attacks are included. The classification results of the class association rules extracted by Fuzzy GNP with simple probabilistic node transition and accurate probabilistic node transition are shown in Table 4.4 and Table 4.5, respectively. In the tables, $Normal(C)$, $Misuse(C)$ and $Anomaly(C)$ indicate the number of normal, misuse intrusions and anomaly intrusions labeled by the hybrid classifier, respectively, while $Normal(A)$, $Misuse(A)$ and $Anomaly(A)$ indicate the actual number of normal, misuse intrusions and anomaly intrusions, respectively.

From the tables, DR, Accuracy, PFR and NFR of Fuzzy GNP with simple proba-

Table 4.5: Classification results of Fuzzy GNP with accurate probabilistic node transitions

	$Normal(C)$	$Misuse(C)$	$Anomaly(C)$	$Total$
$Normal(A)$	1923	7	0	1930
$Misuse(A)$	7	7451	430	7888
$Anomaly(A)$	121	2	59	182
$Total$	2051	7460	489	10000

bilistic node transition are calculated as

$$DR = (1928 + 7444 + 45 + 7 + 278)/10000 = 97.02\% \tag{4.10}$$

$$Accuracy = (1928 + 7444 + 45)/10000 = 94.17\% \tag{4.11}$$

$$PFR = (1 + 1)/1930 = 0.10\% \tag{4.12}$$

$$NFR = (166 + 130)/(7888 + 182) = 3.67\% \tag{4.13}$$

Those of Fuzzy GNP with accurate probabilistic node transition are calculated as

$$DR = (1923 + 7451 + 59 + 2 + 430)/10000 = 98.65\% \tag{4.14}$$

$$Accuracy = (1923 + 7451 + 59)/10000 = 94.33\% \tag{4.15}$$

$$PFR = (1 + 1)/1930 = 0.36\% \tag{4.16}$$

$$NFR = (7 + 121)/(7888 + 182) = 1.59\% \tag{4.17}$$

Comparing Fuzzy GNP with simple probabilistic node transition, the accurate probabilistic node transition one shows better DR, Accuracy and NFR. Although PFR is higher than that in the case of simple probabilistic node transition, the difference between two cases is very slight. The accurate probabilistic node transition contributes to the better performance since it has more precise probabilistic natures than the simple one.

4.5.2 Comparison of Fuzzy GNP with GNP on Classification Performances

Table 4.6: Classification results of GNP

	$Normal(C)$	$Misuse(C)$	$Anomaly(C)$	$Total$
$Normal(A)$	1835	1	94	1930
$Misuse(A)$	78	7324	486	7888
$Anomaly(A)$	135	7	40	182
$Total$	2048	7332	620	10000

The classification performance of GNP is also compared with that of Fuzzy GNP. From Table 4.6, DR, Accuracy, PFR and NFR are calculated as

$$DR = (1835 + 7324 + 40 + 7 + 486)/10000 = 96.92\% \tag{4.18}$$

$$Accuracy = (1835 + 7324 + 40)/10000 = 91.99\% \tag{4.19}$$

$$PFR = (1 + 94)/1930 = 4.92\% \tag{4.20}$$

$$NFR = (78 + 135)/(7888 + 182) = 2.64\% \tag{4.21}$$

Fig. 4.11 and Fig. 4.12 show the comparisons of the classification results among GNP and Fuzzy GNP with two different kinds of node transition, respectively. As higher DR and Accuracy are better, while lower PFR and NFR are better. Therefore, from the comparisons between Fuzzy GNP and GNP, both Fuzzy GNPs with accurate node transition and simple node transition have higher DR and Accuracy than those of GNP. Meanwhile, Fuzzy GNP decreases PFR and NFR comparing with GNP. Therefore, the proposed Fuzzy GNP has better detection performance. Fuzzy GNP has the ability to extract more useful class association rules than GNP.

On the other hand, the accurate probabilistic node transition presents more precise probabilistic nature than simple probabilistic method. From the figures, DR, Accuracy and NFR except PFR are better than the simple one. PFR is only 0.36%.

In addition, the performance of the proposed Fuzzy GNP is compared with some other machine learning techniques for intrusion detection. All the simulation results in

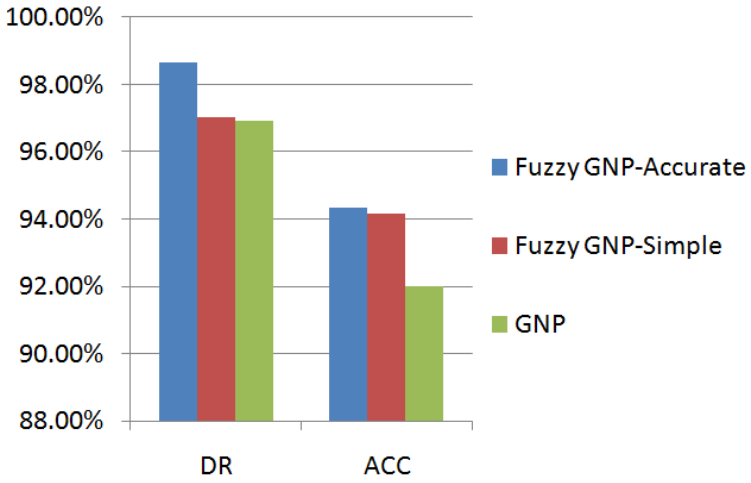


Figure 4.11: Comparing the classification results of Fuzzy GNP and GNP on DR and Accuracy

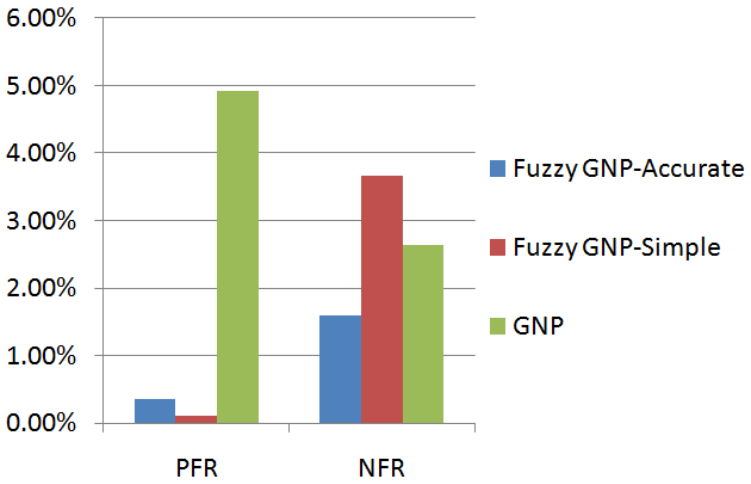


Figure 4.12: Comparing the classification results of Fuzzy GNP and GNP on PFR and NFR

Table 4.7 are conducted on KDD99Cup Dataset for fair comparisons(97). It is found from Table 4.7 that the proposed method outperforms the conventional machine learning techniques for intrusion detection in terms of detection rate. In Table 4.7, $APNT$ represents accurate probabilistic node transition and $SPNT$ means simple probabilistic node transition.

Table 4.7: Performance of comparisons among the intelligent methods

Technique	*Detection Rate*(%)
Fuzzy GNP with APNT	98.65
Fuzzy GNP with SPNT	97.02
GNP	96.92
*C*4.5	95.0
Support Vector Machine	95.5
Muilti Layer Perception	94.5
$K - Means\ Clustering$	65.0
Hidden Markov Model(*HMM*)	79.0
$C4.5 + Hybrid\ Neural\ Networks$	93.3
Genetic Programming	91.0
$K - Nearest\ Neighbor$	92.0
$Neural\ Networks + PCA$	92.2

4.6 Conclusions

In this chapter, Fuzzy GNP is proposed to extract class association rules from network connection data. In addition, the average matching degree is correspondingly modified considering fuzzy part and non-fuzzy part of one rule.

Fuzzy set theory is integrated into GNP, aiming at dealing with the fuzziness in the class association rules of intrusion detection problem and the sharp boundary problem in crisp discretization of GNP-based class association rule mining. Each generation gets the fuzzy values for each continuous attribute and the parameters of the fuzzy membership functions are evolved by non-uniform mutation in order to perform more global search in early stages and local search in later stages. Besides, the proposed accurate probabilistic node transition can contribute to the diversified class association rules. By this way, both quality and quantity are improved by Fuzzy GNP with probabilistic node transition. The simulation results on KDD Cup 1999 indicate that the proposed method can get good detection performance. And the accurate probabilistic node transition can extract more useful rules than simple probabilistic node transition and has better detection ability owing to its precise probabilistic nature. It is also showed that the proposed Fuzzy GNP is better than other conventional machine learning approaches for intrusion detection in terms of Detection Rate.

However, it still has much space to improve. From the simulation analysis, the anomaly intrusions are hard to detect in the hybrid framework of intrusion detection system because of its natures of new and diversity. Therefore, an effective classifier is needed to accurately identify normal, misuse and anomaly intrusions.

5

Classification for Intrusion Detection System using Distance Approach

Building an accurate and efficient classifier is one of the essential tasks of data mining and machine learning research. As stated in (63), classification generally maps a data into one of several predefined categories. An ideal approach in intrusion detection would be to learn a classifier from gathered normal and intrusion data, then label or predict new unseen data as the normal class or intrusion class.

In addition, the two-stage rule pruning method alleviate the overlapping problem by pruning the redundant and irrelevant rules from the rule pool. An efficient classification method is needed to deal with the overlapping part since the rule pruning cannot thoroughly solve this problem. In order to enhance the detection ability of IDS, Chapter 6 and Chapter 7 will focus on the classification algorithms of intrusion detection which can better use the class association rules extracted by Fuzzy GNP.

5.1 Introduction

Since efficient classification algorithms are extremely important for intrusion detection, a large number of studies has been conducted. K-Nearest neighbor(KNN) is an extremely simple yet surprisingly effective method for a classification. Its advantages stem from the fact that its decision surface are nonlinear with only a single integer parameter. More importantly, these advantages do not cause the over-fitting(103)(28), and it is not restricted to any specific data distribution.

In this chapter, a novel distance-based classification method is proposed, which originates from K-Nearest Neighbor (KNN) decision rule, where, the multiple feature space is projected into a two-dimensional space describing the degree that the connection data belong to normal or misuse intrusion based on the average matching degree. Then, K-Closest neighbor classifier is employed to categorize each new data into either normal or misuse intrusion. However, usually anomaly connection data have been mixed into normal or misuse intrusion. Considering the above, a multiple centroid-based modification is introduced in order to improve the detection performance. In the proposed method, the centroids of anomaly intrusion data are defined by the centroids of normal data and misuse intrusion data. The importance of the proposed method is that misuse intrusion data also have the contribution to detecting anomaly intrusion data. The main features of Distance-based classification are summarized as follows.

1. It is a non-parametric approach, where only the number of the closest neighbors should be determined. Whereas, the simulations on different numbers of the closest neighbors indicate that the detection ability is not so sensitive to this number.

2. The nature of anomaly intrusion is take into account by making full use of the information from normal and misuse intrusion connection data. Therefore, the centroids of different classes are proposed to make the classification.

The rest of this chapter is organized as follows. The motivations of this chapter is presented in Section 5.2. Section 5.3 describes the data set used in this chapter. A new Distance-based classification method is introduced in Section 5.4. Simulations are shown in Section 5.5. Finally, we conclude the summary in Section 5.6.

5.2 Motivations

The two-stage rule pruning method can reduce the influence of the redundant and irrelevant rules to alleviate the overlapping problem, which occurs in the two-dimensional average matching degree space. However, it cannot fully solve the overlapping in the

case that two connection data have similar behaviors, especially for some anomaly intrusions similar to normal behaviors. Therefore, an efficient classification is needed to deal with such an overlapping problem.

On the other hand, in hybrid classifier, the mean and standard deviation model shows that the classification boundary is linear. Intrusion detection is a complex classification problem. It contains a wide variety of network behaviors. Moreover, it is a synthesis of multi-class classification and anomaly identification. A linear model is difficult to get better results because intrusion detection is a non-linear problem actually. Classification boundary is not linear by finding the nearest neighbors for each new data. And it has no unique model for classification.

In addition, anomaly intrusions are difficult to detect. This is an important reason that accuracy is not good enough. From the other aspect, it is vital to identify anomaly intrusions from normal and misuse intrusions. Traditional anomaly intrusion detection techniques assume that there is only "normal" class and any data that does not belong to the normal class is an anomaly class data. But in reality, the new connection data has the possibility belonging to misuse intrusions. And identifying anomaly intrusions exactly is essential to analyze new intrusion class for further protection of computers and networks. Because there is no information about anomaly intrusions. How to make full use of known information is critical to improve the detection ability. Using detail information to detect the extract class of the data is not good, because it will be biased to part of the data. Therefore, overall information is used to define the general regions of anomaly intrusions.

5.3 Data Description

NSL-KDD(104)(105) is used to conduct the simulations as the subset of KDD Cup 1999 data set. This data set is important to evaluate the intrusion detection systems especially anomaly intrusions. NSL-KDD data set avoids two problems of KDD Cup 1999 data set for network-based anomaly detection. The first problem is the huge number of redundant records. Analyzing KDD Cup 1999, 78% and 75% of the records are duplicated in the training set and testing set, respectively. Too many redundant records may cause the algorithms to be biased towards the more frequent records, and thus prevent it from learning unfrequent records. The existence of these repeated

records in the testing set, may cause the evaluation results to be biased by the methods which have better detection rates on the frequent records. The other problem is the level of difficulty. In general, the typical approach for performing anomaly detection using the KDD Cup 1999 data set is to employ a machine learning algorithm to learn the general behavior of the data set in order to be able to differentiate between normal and malicious activities. In order to improve the level of difficulty in testing set, NSL-KDD only select the records which are difficult to be identified. To solve the two issues, NSL-KDD removed all the repeated records in the entire KDD Cup 1999 training and testing set and kept only one copy of each record.

Same with KDD Cup 1999 data set, each record of NSL-KDD data set has 41 features. All the data in NSL-KDD are labeled and classified into normal or 24 training attack types, with additional 14 types in the test data only.

5.4 Classification using Distance Approach

In the distance-based classification model, the basic idea is to represent data d as a combination of the average matching degree with the rules in class k, $k \in C = \{1, 2, ..., L\}$, which is the coordinate of $(m_1(d), m_2(d), ..., m_L(d))$ in a L-dimensional space calculated by Eq.(3). In case of Intrusion Detection Systems, there are 2 classes: one is normal and the other is misuse intrusion. Thus, the coordinate of data d is $(m_N(d), m_I(d))$.

Building phase of average matching degree space In order to build the distance-based classification model, all training data should be mapped into the L-dimensional average matching degree space. As to IDS, normal data is included in the training data as well as misuse intrusion data. For this reason, each training data d should be represented as the coordinate of $(m_N(d), m_I(d))$ in the 2-dimensional average matching degree space.

Fig. 5.1 shows an example of the distributions of both normal and misuse intrusion training data in the 2-dimensional average matching degree space. However, in real applications, we should point out that the distribution of the average matching degree calculated by training data is scattered and also the distributions of these two classes are overlapped to some extent as shown in Fig. 5.2. This is one of reasons that leads to inaccuracy in the classification phase.

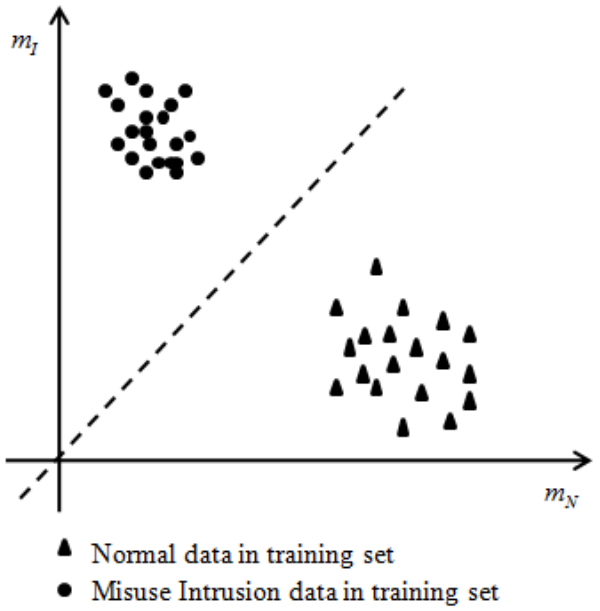


Figure 5.1: An example of 2-dimensional average matching degree space for IDS

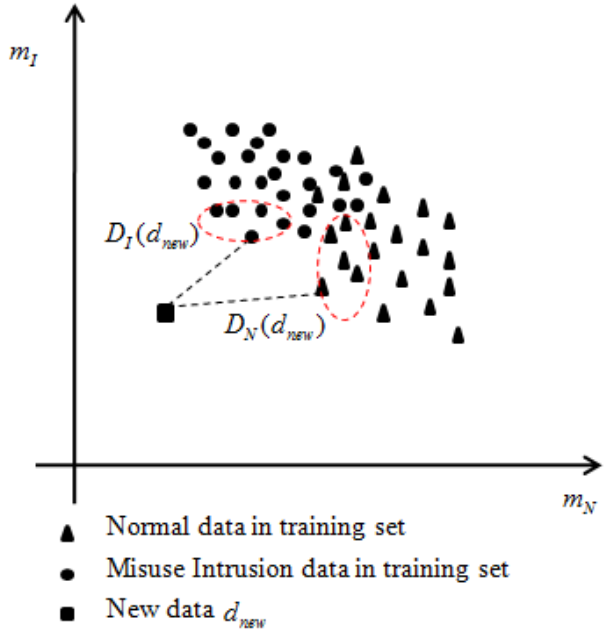


Figure 5.2: Overlapping in 2-dimensional average matching degree space for IDS

Fuzzy GNP has the ability of extracting a huge number of class association rules for classification. However, too many rules usually contain many redundant, irrelevant

and obvious information, which has the negative influence on distinguishing normal and intrusion. In this case, the utilization of such rules leads to the overlapping in the 2-dimensional average matching degree space.

Therefore, the two-stage rule pruning method is used to alleviate the overlapping problem, where no domain knowledge is needed for pruning as in chapter 3. After pruning, the conflict rules are eliminated and the general rules are reserved.

Classification phase in average matching degree space

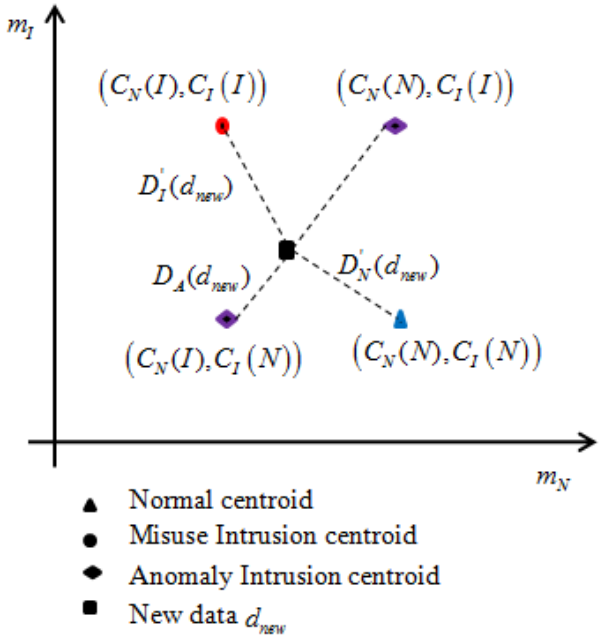


Figure 5.3: Distance-based classification model

When new data d_{new} is observed, d_{new} is also represented as the coordinate of $(m_N(d_{new}), m_I(d_{new}))$. The Euclidean distance between new data d_{new} and training data d in class k in the 2-dimensional average matching degree space, denoted by $D_k(d_{new})$, can be defined as follows,

$$D_k(d_{new}) = \frac{1}{|D_{K-Closest}(k)|} \sum_{d \in D_{K-Closest}(k)} D(d_{new}, d), \tag{5.1}$$

where, $D(d_{new}, d)$ is the distance between new data d_{new} and training data d in the 2-dimensional average matching degree space, $D_{K-Closest}(k)$ is the set of suffixes of K-Closest training data in class k in the 2-dimensional average matching degree space

to new data d_{new}. It is reasonable to tell that new data d_{new} is more likely to be normal data or misuse intrusion data by comparing $D_N(d_{new})$ and $D_I(d_{new})$. That is, new data d_{new} is labeled as the class of the smaller distance between $D_N(d_{new})$ and $D_I(d_{new})$. As Eq. (5) shows, K-closest neighbor uses the average value of the distances of data d_{new} to its K closest training points. The application of K-Closest neighbor method can deal with the overlapping problem to some extent by using the important information on the training data for classification.

In addition, anomaly intrusions should be considered to improve the performance of intrusion detection, because it is dangerous that new attacks are easily regarded as normal connections. Thus, it is appropriate to suppose some points as the centroids of anomaly intrusions. However, we have no information about such anomaly intrusions. So, we should analyze the available information from the normal and misuse intrusion data in the training set to solve this problem.

Based on the given information about normal data and misuse intrusion data, the centroid point $(C_N(N), C_I(N))$ of normal training data, called normal centroid, can be calculated using Eqs.(5.2)∼(5.3).

$$C_N(N) = \frac{\sum_{d \in D_{Train}(normal)} m_N(d)}{|D_{Train}(normal)|}, \tag{5.2}$$

$$C_I(N) = \frac{\sum_{d \in D_{Train}(normal)} m_I(d)}{|D_{Train}(normal)|}, \tag{5.3}$$

where $D_{Train}(normal)$ is the set of suffixes of normal training data. While the centroid point $(C_N(I), C_I(I))$ of misuse intrusion training data, called misuse intrusion centroid, can also be calculated in the same way. Then, anomaly centroid points are manually set by using the coordinates of normal centroid and misuse intrusion centroid, i.e., $(C_N(I), C_I(N))$ and $(C_N(N), C_I(I))$.

Once the four centroids are determined, the distances of new data d_{new} to normal or misuse intrusion can be recalculated by $D'_N(d_{new})$ or $D'_I(d_{new})$. The distance of new data d_{new} to anomaly intrusion centroids, denoted by $D_A(d_{new})$, is also calculated. Fig. 5.3 shows the basic idea to decide the centroids for anomaly intrusion data, where $D'_N(d_{new})$ and $D'_I(d_{new})$ are calculated by Eqs.(5.4)∼(5.5) using normal centroid and misuse intrusion centroid, meanwhile $D_A(d_{new})$ is calculated by Eq.(5.6) using anomaly

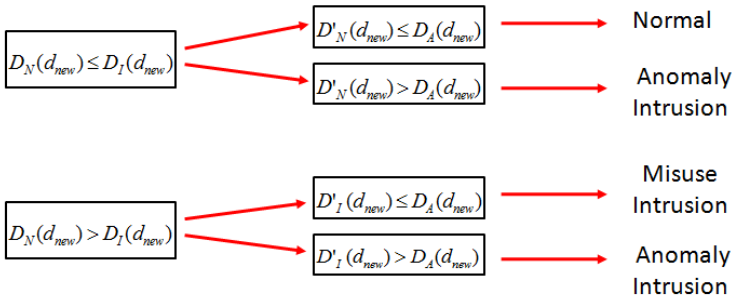


Figure 5.4: Classification procedure

intrusion centroids. Since the anomaly intrusions usually pretend to be normal behaviors to attack the computers or systems, they are similar with normal behaviors to some extent. However, they are new intrusions in nature. Thereby, they have some features of misuse intrusions, but do not equal to the known intrusions completely. Therefore, its reasonable to assume that anomaly intrusions are like neither misuse intrusion nor normal behaviors, or like both misuse intrusion and normal behaviors. The general procedure for intrusion detection is shown in Fig. 5.4.

$$D'_N(d_{new}) = \sqrt{(m_N(d_{new}) - C_N(N))^2 + (m_I(d_{new}) - C_I(N))^2} \tag{5.4}$$

$$D'_I(d_{new}) = \sqrt{(m_N(d_{new}) - C_N(I))^2 + (m_I(d_{new}) - C_I(I))^2} \tag{5.5}$$

$$\begin{aligned} D_A(d_{new}) = min\{&\sqrt{(m_N(d_{new}) - C_N(I))^2 + (m_I(d_{new}) - C_I(N))^2}, \\ &\sqrt{(m_N(d_{new}) - C_N(N))^2 + (m_I(d_{new}) - C_I(I))^2}\} \end{aligned} \tag{5.6}$$

5.5 Simulations

To evaluate the proposed method, the simulations are conducted mainly using the data set of NSL-KDD in addition to KDDCup 1999 data set.

Since NSL-KDD data set is not biased toward frequent records, the training set in NSL-KDD is used to extract rules. We randomly selected $4,000$ data for the training data set from NSL-KDD, which has $2,000$ normal data and $2,000$ misuse intrusion data consisting of $1,500$ with neptune type and 500 with smurf type.

Table 5.1: Parameters of Fuzzy GNP-based class association rule mining

Population Size	120
Generation	1000
Processing Node	10
Judgment Node	100
Crossover Rate	1/5
Mutation Rate P_{m1}	1/3
Mutation Rate P_{m2}	1/3
χ^2_{min}	6.64
$support_{min}$ (N)	0.1
$support_{min}$ (I)	0.075
$confidence_{min}$	0.8

For rule extraction, Fuzzy GNP-based class association rule mining is applied and its parameters are listed in Table 5.1, where, $support_{min}$ (N) means the minimum support value to select interesting normal rules, and $support_{min}$ (I) means the minimum support value to select interesting misuse intrusion rules. After $1,000$ generations of evolution, Fuzzy GNPs extract $50,324$ rules consisting of $15,149$ normal rules, $25,022$ neptune misuse intrusion rules and $10,153$ smurf misuse intrusion rules. Fig. 5.5 shows the total number of accumulated rules extracted vs. generation.

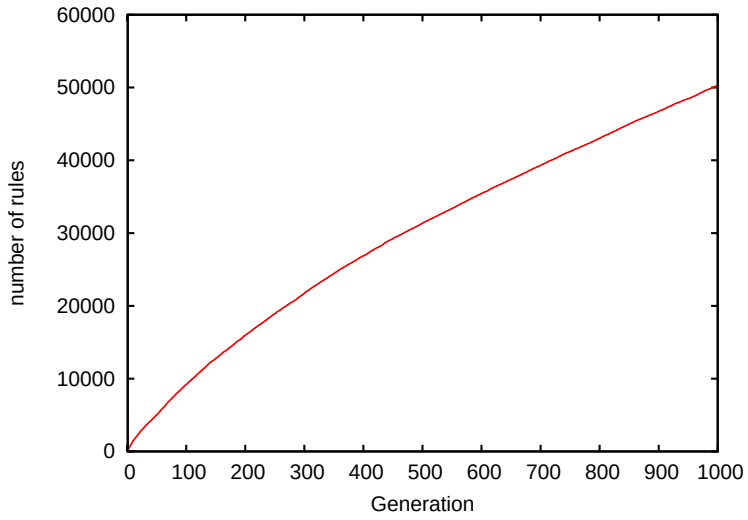


Figure 5.5: Number of accumulated rules extracted vs. generation

Table 5.2: Results of distance-based classification on KDD Cup 1999

	Probe	*U2R*	*R2L*	*DoS*	*neptune & smurf*
Number of test data	4,116	228	12,189	229,851	222,092
DR	94.83%	75.88%	70.73%	97.54%	99.98%

5.5.1 Performances of Distance-based Classifier

As the number of rules is too large to build the classifier, the two-stage rule pruning mechanism is implemented to pick up useful rules and to reduce the influence of redundant and irrelevant rules in the rule pool. As a result, there remain only 432 normal rules, 416 misuse intrusion rules with *neptune* type and 152 misuse intrusion rules with *smurf* type after pruning. Then, the average matching degrees of each training data with normal rules or misuse intrusion rules are mapped into the two dimensional space.

Even though many redundant and irrelevant rules are generated in the rule pool, the class association rule mining is effective and efficient. The reasons are as follows.

In class association rule mining, the values of the *support*, *confidence* and χ^2 are main criteria to evaluate whether a rule is important or not. If the minimum values of the *support*, *confidence* and χ^2 are set at high values, the number of the class association rules extracted becomes very small. But, it is not a good approach. Generally speaking, many rules with lower criteria should be extracted first, then some of them should be pruned in terms of obtaining the better performance for classification, because the rules with low criteria might contribute to the better performance.

In this section, two simulations are performed. Firstly, the test data from original KDD Cup 1999 data set are used to check DR of the proposed method on the four kinds of attacks. Table 5.2 shows DR of each kind of attacks with $K = 30$. In addition to the four kinds of attacks, the case is also listed where only *neptune* type and *smurf* type exist in the group of *DoS* intrusion. From Table 5.2, DR on these two types reaches 99.98%. Even those types do not exist in the training data, the proposed classifier can detect the intrusions well.

Next, the $KDDTest^+$ data in NSL-KDD data set is considered to evaluate the effectiveness of the distance-based classifier with $K = 30$. DR, *Accuracy*, PFR and NFR are used to evaluate the performance of the proposed classifier in this experiment.

Table 5.3: Results of distance-based classification on NSL-KDD

	$Normal(C)$	$Misuse(C)$	$Anomaly(C)$	$Total$
$Normal(A)$	8,080	168	1,463	9,711
$Misuse(A)$	0	5,301	21	5,322
$Anomaly(A)$	1,910	1,010	4,591	7,511
$Total$	9,990	6,479	6,075	22,544

In this data set, each connection record is different from each other, meanwhile, normal, misuse intrusion and anomaly intrusion are all included.

The classification results of the proposed method are shown in Table 5.3. $Normal(C)$, $Misuse(C)$ and $Anomaly(C)$ indicate the number of normal, misuse intrusions and anomaly intrusions labeled by the Distance-based classifier, respectively, while $Normal(A)$, $Misuse(A)$ and $Anomaly(A)$ indicate the actual number of normal, misuse intrusions and anomaly intrusions, respectively. From Table 5.3, it is obvious that most of anomaly intrusion data can be detected by this classifier, which means it has better detection ability on anomaly intrusions.

According to Table 5.3, DR, Accuracy, PFR and NFR are calculated as shown in Eq.(5.7)∼(5.10).

$$DR = (8,080 + 5,301 + 4,591 + 1,010 + 21)/22,544 = 84.29\% \tag{5.7}$$

$$Accuracy = (8,080 + 5,301 + 4,591)/22,544 = 79.72\% \tag{5.8}$$

$$PFR = (168 + 1,463)/9,711 = 16.80\% \tag{5.9}$$

$$NFR = (1,910 + 0)/(5,322 + 7,511) = 14.88\% \tag{5.10}$$

From Eq.(5.7)∼(5.10), even on the $KDDTest^{+}$ data set which is difficult to detect, the proposed classifier can get good performance. Since the results are obtained by conducting simulations using the NSL-KDD testing data set, the testing data show the high difficulty level in different types of attacks. Though 14.88% seems a little bit high, 4,591 in Table 4 represents that many of the anomaly intrusions can be detected accurately. The proposed method is efficient especially for anomaly intrusions because

the anomaly intrusions are usually similar to normal data and too difficult to detect. The proposed classification method provides more accurate information to the classifier in terms of adding the centroids for anomaly intrusions.

Fig. 5.6 and Fig. 5.7 show the comparisons of the proposed classifier and hybrid classifier, where Fig. 5.6 shows the comparisons of DR and Accuracy, while Fig. 5.7 shows the comparisons of PFR and NFR. The set of rules used in hybrid classifier is also extracted by Fuzzy GNP and has been pruned by two-stage rule pruning method.

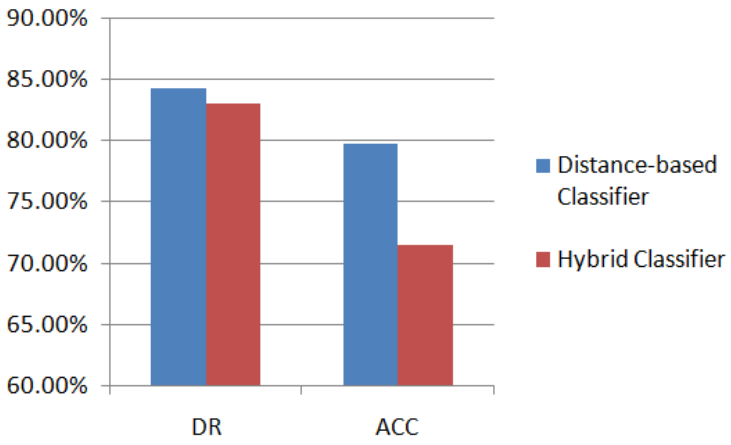


Figure 5.6: DR and ACC comparisons between distance-based classifier and hybrid classifier

Obviously, DR and $Accuracy$ of the Distance-based classifier is higher than those of hybrid classifier, in addition, PFR of the proposed classifier is lower than that of hybrid classifier. Only NFR is a little bit higher than hybrid classifier. Consequently, the performance of the Distance-based classifier outperforms hybrid classifier. And the reason for better performance of the proposed classifier is that it provides more accurate information to the classifier in terms of adding the centroids for anomaly intrusions.

On the other hand, since the distance-based classifier is a non-parametric method, the number K of nearest neighbors is the only factor which should be pre-determined. Therefore, we also checked the influence of this factor on the performance of the proposed method. DR, $Accuracy$, PFR and NFR are not so sensitive to K as shown in Table 5.4.

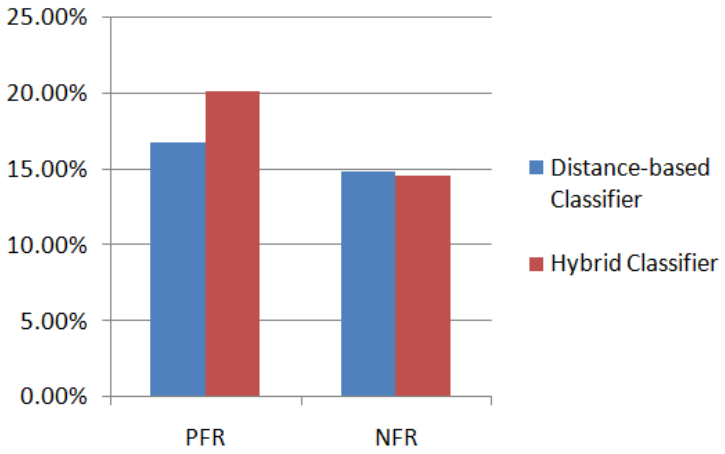


Figure 5.7: PFR and NFR comparisons between distance-based classifier and hybrid classifier

Table 5.4: Influence by the number of K-nearest neighbors

	$K = 5$	$K = 10$	$K = 20$	$K = 30$	$K = 50$	$K = 100$	$K = 200$	$K = 300$
DR	83.94%	84.21%	84.22%	84.29%	84.21%	84.21%	84.21%	83.58%
Acc	76.21%	77.17%	78.48%	79.72%	78.99%	78.56%	78.26%	77.31%
PFR	17.25%	17.25%	16.90%	16.80%	16.97%	17.07%	17.44%	17.52%
NFR	15.16%	14.92%	14.94%	14.88%	14.90%	14.82%	14.77%	15.59%

In conclusion, K-nearest neighbors is first used to label the data temporarily. It is necessary because we should know the data is like either normal or intrusion. Step 2 uses the centroids of normal and intrusion to decide the centroids of anomaly. Because most of anomaly intrusions are like normal or intrusion and not like normal or intrusion, too. That means the distance between anomaly intrusion and normal(or intrusion) is not so near and not so far. So, this is the important reason that detection performance is improved in this method.

5.5.2 Comparisons with Other Methods

The proposed method belongs to the supervised learning for classification. Therefore, the proposed classifier is only compared with two classical supervised methods, where Multi-Layer Perceptron is a basic neural network method, while Support Vector Machine is an active machine learning method in recent years for solving a variety of

regression and classification.

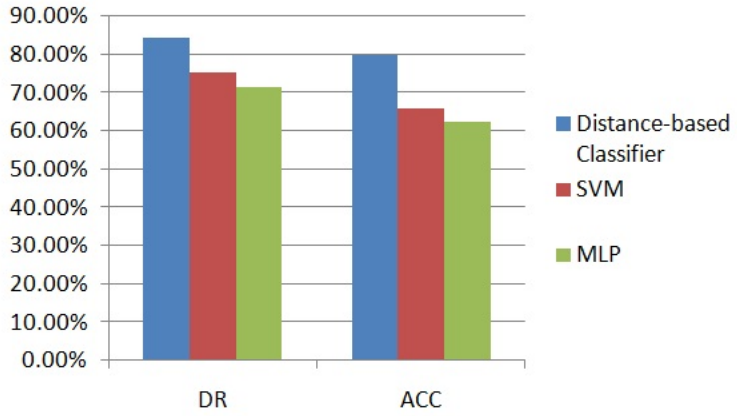


Figure 5.8: DR and ACC comparisons among distance-based classifier, SVM and MLP

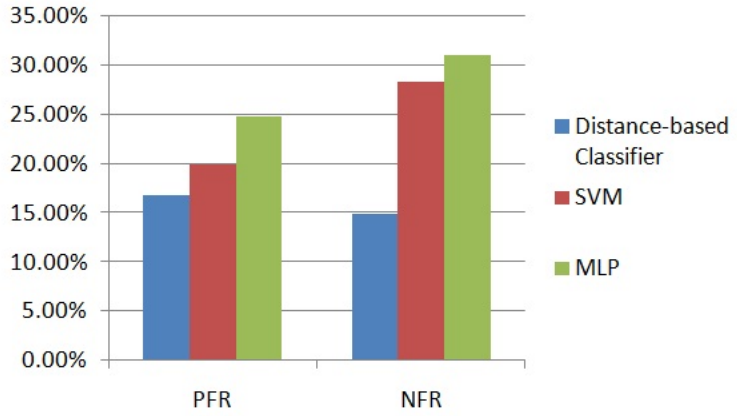


Figure 5.9: PFR and NFR comparisons among distance-based classifier, SVM and MLP

Support Vector Machine(SVM) has been widely used for IDS as a classical patter recognition technique. In this method, the average matching degrees m_N and m_I over all training data are regarded as the inputs of SVM, while the output is the similarity probability of the test data to normal patterns or misuse intrusion patterns. If the test data has higher similarity probability to normal patterns than misuse intrusion patterns by a certain value, we regard it as normal, while if the test data has higher similarity probability to misuse intrusion patterns than normal patterns by a certain value, it is regarded as misuse intrusion. Otherwise, it is classified as anomaly intrusion.

The Libsvm(106) tool is used to perform the simulations with SVM. As the simulation parameters of SVM, C-SVC SVM algorithm is used, i.e., selected the radial basis function as the kernel type and used other relevant default parameters.

Multi Layer Perceptron(MLP) is one of the most commonly used neural networks for classification. The architecture used for MLP consists of four layers, i.e., two inputs, two hidden layers(five nodes in each layer) and one output. The average matching degrees m_N and m_I over all training data are also used as its inputs to train the MLP classifier. Fig. 5.8 and Fig. 5.9 show the comparisons among the Distance-based classification, SVM and MLP in terms of DR, Accuracy, PFR and NFR.

From the aforementioned comparisons, the performance is improved by the distance-based classifier due to that misuse intrusions can be detected accurately, at the same time, the detection of anomaly intrusions is reasonable and effective to most of the anomaly intrusion data. Moreover, the proposed classifier provides more accurate information without parameter tuning, whereas, it is necessary to select appropriate parameters to build the classifier using SVM or MLP.

5.6 Conclusions

In this chapter, a two-step Distance-based classification method has been proposed using the average matching degree of the connection data with rules. To evaluate the performance of the proposed method, both original KDDCup 1999 data set and NSL-KDD data set which removed redundant records from KDD Cup 1999 are simulated to increase the level of difficulty. It has been clarified from simulation results that the proposed Distance-based classifier can get higher DR and Accuracy, and lower PFR and almost similar NFR to the hybrid intrusion detection classifier. In addition, it is found that DR, $Accuracy$, PFR and NFR are not so sensitive to K. The proposed classifier is also compared with other two well-known methods. The results show the proposed one has better performance. It is remarkable that the proposed classifier can detect all the connection data of *neptune* type and $smurf$ type in the test set of KDD Cup 1999.

On the other hand, the detection of anomaly intrusions still has room for improvement. As is known, anomaly intrusions are actually similar to normal or intrusion

behaviors. In order to make full use of normal and misuse intrusion patterns, it is essential to find the exact boundary of normal and misuse intrusion patterns.

6

Classification for Intrusion Detection System using Gaussian Functions

Anomaly detection is an important problem that has been studied within diverse research areas and application domains, especially for computer security(14). Finding the hardest-to-detect anomalies is the most critical task in intrusion detection.

In the hybrid framework of the intrusion detection system, it is easy to get good performance when identifying the misuse intrusions from normal data. Whereas, anomaly intrusions are usually difficult to identify because of its no patterns. Traditional detection method of anomaly intrusions relies on normal patterns. However, in reality, the behaviors of normal connection data are too diverse to gather completely. Therefore, if the types of both normal behaviors and misuse intrusion behaviors are considered and the boundary of each type of behaviors is found, then it becomes simple to identify a new connection as normal or intrusion.

6.1 Introduction

It is crucial to adopt an appropriate classification approach for intrusion detection systems. Probabilistic classification proposed in (79) assumes that the normal class and misuse intrusion class are independent to estimate the two one dimensional probability density functions which represent the distribution of the data of the normal class and misuse intrusion class, respectively. However, in the field of intrusion detection, the

probability density functions of the normal class and misuse intrusion class are usually correlated.

In distance-based classification(107), known information of normal and intrusion is used to determine the possible regions of anomaly intrusions. Centroids of anomaly intrusions are defined by normal centroid and intrusion centroid. However, anomaly intrusions are still difficult to distinguish because some of them are close too much to normal or known intrusions. Therefore, it is feasible to identify an anomaly intrusion if the exact boundaries of normal and known intrusions can be found.

In this chapter, a new approach is proposed to find such the boundaries of normal and known intrusions. In order to make full use of known information about normal and known intrusions, it is essential to group the similar data into the same cluster, which means they have similar behaviors. Then, the problem becomes finding exact boundary for each cluster. This method intends to solve two points. One is the appropriate number of clusters. The other is the determination of the boundary for each cluster.

The advantages of the new proposed classification approach are summarized in the following.

1. Both normal and misuse intrusion contain more than one type of behaviors. The clustering is used to gather similar patterns in one cluster automatically.

2. A new clustering method is used by dividing the average matching degree space into many blocks. Each block corresponds to a cluster.

3. Gaussian function is used to decide the boundary of the cluster. Each cluster has its own Gaussian function. All of the Gaussian functions are different with each other.

4. The center of the cluster equals to the center of its Gaussian function. The boundary of the cluster is decided by GA considering its classification performance.

The rest of this chapter is organized as follows. Section 6.2 gives the motivations of the proposed method. Section 6.3 describes the new classification method using

clustering and Gaussian functions. Simulations are shown in Section 6.4. Finally, the summary is concluded in Section 6.5.

6.2 Motivations

Classification represents a widely studied domain. Currently, in most of the machine learning approaches, solutions still generally take the form of a single classifier per class. However, there are too many patterns existing in normal and misuse intrusion connection data for intrusion detection. Even though no classifier can model too many behaviors, it is appropriate to build a classifier for a cluster of similar behaviors. Hence, the two issues come out. One is how to decide the clusters for the behaviors of the training space. The other is what kind of model is used to build the classifier. Here, more attention should be paid to one phenomenon. When the distance-based classification approach is used, the detection performance showed that the anomaly intrusion connection data was easily identified as normal or misuse intrusion, while the part of normal connection data was identified as anomaly intrusion, which resulted in high PFR. The solution of the misclassified parts is an important factor to improve detection performance. Where are the misclassified parts from? Because all normal data are considered as one cluster and all intrusion data are considered as one cluster. It is not exact because both normal data and intrusion data belong to more than one cluster.

Each cluster is supposed to obey the gaussian distribution since gaussian distribution is a common hypothesis in all the data analysis. Moreover, Gaussian function conforms to requirement of this research. The shape of it can be controlled by the parameters of Gaussian function. The location of the data in average matching degree space can be mapped to the score which means how much the data belongs to this Gaussian function.

6.3 Classification using Gaussian Functions

Different from many other classification applications, IDS is special since it aims at detecting not only the connections with known patterns, but also those without attack signatures. The accurate detection of anomaly intrusions can reduce NFR and PFR, and improve Accuracy of IDS. However, due to the lack of the patterns of anomaly intrusions, it is difficult to detect anomaly intrusions from the mixture of normal and

misuse intrusions. In order to keep IDS work well, the precise detection of anomaly intrusions is essential for extracting useful patterns from brand new intrusions.

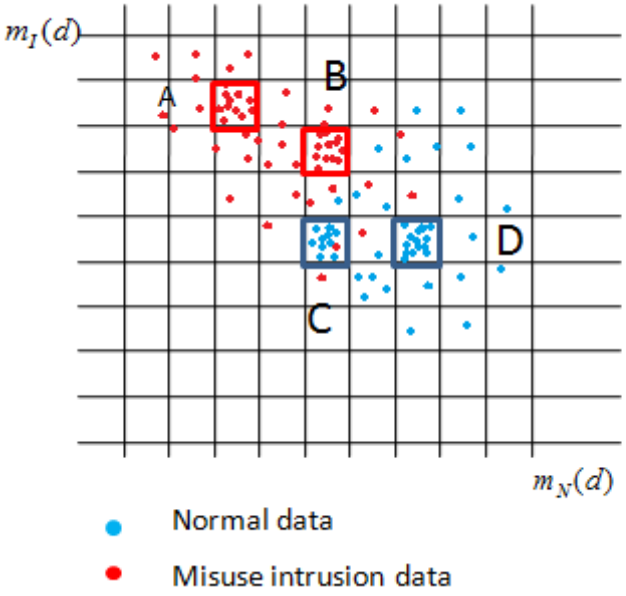


Figure 6.1: Determination of clusters

6.3.1 Clustering Network Behaviors

In this section, we will explain how to determine each cluster. Different from K-means clustering approach, the proposed approach uses the distribution of the average matching degrees to define the clusters.

The basic idea is to segment the two-dimensional space of the average matching degree corresponding to the normal class ($m_N(d)$) and misuse intrusion class ($m_I(d)$) into many blocks as shown in Fig. 6.1. Naturally, the average matching degrees fall into different blocks by their coordinates. It is shown from Fig. 6.1 that the points of the average matching degrees of the training data in some blocks are dispersed and those of other blocks are crowded. Then, the crowded ones are regarded as clusters. In Fig. 6.1, blue points represent the average matching degrees labeled as normal, while red ones represent those labeled as misuse intrusion. Block A and B contain more points of misuse intrusion than the minimum number of points in a block, while block C and D

contain more normal points than the minimum number of points in a block. Therefore, A, B, C and D correspond to the clusters.

6.3.2 Classification Model based on Gaussian Functions

Each cluster has a single Gaussian function, whose center equals the center of the corresponding block. In the proposed method, the following two-dimensional Gaussian function is used.

$$f(m_N, m_I) = Ae^{-(a(m_N-\mu_N)^2+b(m_N-\mu_N)(m_I-\mu_I)+c(m_I-\mu_I)^2)}, \tag{6.1}$$

where, coefficient A is an amplitude, (μ_N, μ_I) is the center of the cluster corresponding to the coordinates of normal and misuse intrusion in the two dimensional average matching degree space, a, b and c are the parameters of the Gaussian function to adjust. In this application, A is set at 1. Therefore, the values of Gaussian functions range from 0 to 1.

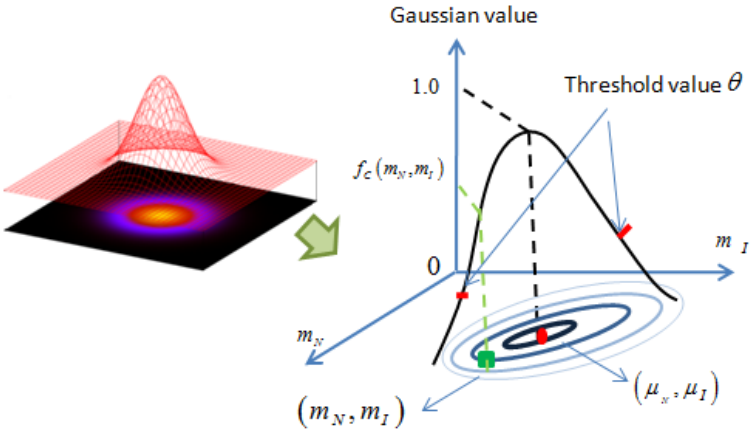


Figure 6.2: A single Gaussian function

Fig. 6.2 shows a single Gaussian function. If the average matching degree (m_N, m_I) of a new connection data is close to the center of the cluster, the corresponding Gaussian function can have a high value. Then, threshold θ is used to evaluate how much a new connection data is close to the class of the corresponding Gaussian function. $f_c(m_N, m_I)$ in Fig. 6.2 denotes the value of a Gaussian function for a new connection data. If $f_c(m_N, m_I)$ is larger than threshold θ, a new connection data has the same class label as that of the corresponding Gaussian function.

Actually, both the normal class and misuse intrusion class contain more than one Gaussian function. $GV(m_N, m_I)$ is calculated by the following equation,

$$GV(m_N, m_I) = \max\{f_1(m_N, m_I), ..., f_c(m_N, m_I), ..., f_C(m_N, m_I)\}, \tag{6.2}$$

where, $c \in C$ is the cluster number and its set for normal or misuse intrusions. Then, $GV(m_N, m_I)$ is actually like a score which is used to determine the class of a new connection data as shown in Fig. 6.3, where GV_N and GV_I are the gaussian value for normal and misuse intrusion, respectively. Step 1 identifies whether a new connection data belongs to normal or not. In step 2, the class of a new connection data can be determined using threshold θ. In this chapter, threshold θ of normal and misuse intrusions equals to 0.99.

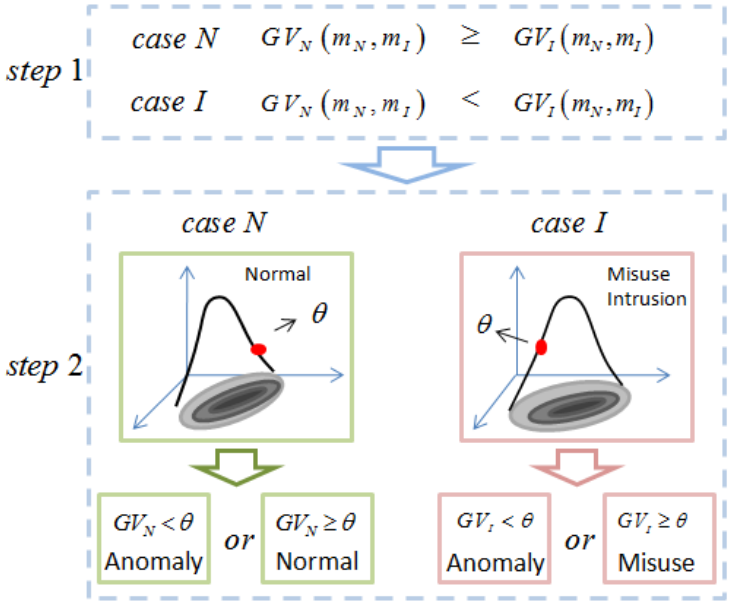


Figure 6.3: Classification procedure

Though the centers of Gaussian functions are determined, another problem is to determine the shape of Gaussian functions. In the two-dimensional average matching degree space, parameter a, b and c of Gaussian functions are used to determine the appropriate boundary of each cluster.

6.3.3 Boundary Estimation by GA

In the proposed method, GA is used to find the optimal a, b and c of each Gaussian function. In order to balance the exploitation and exploration and alleviate the premature convergence, non-uniform mutation and blend crossover are used in GA. The following fitness function is used to evaluate the performance of each individual.

$$fitness = \alpha * ACC - \beta * PFR - \gamma * NFR, \tag{6.3}$$

where, ACC is the classification accuracy by the set of Gaussian functions. PFR is positive false rate and NFR is negative false rate. The individuals are ordered by their fitness values. Better individuals are selected to do crossover and mutation by tournament selection. As crossover operator, blend crossover $BLX - \alpha$ is used. As usual, individual x^1 and x^2 are selected as parents. Each element x_i^c of individual x^c ($c \in \{1, 2\}$) of the offspring is chosen from the interval $[X_i^1, X_i^2]$ with the crossover rate.

$$X_i^1 = \min(x_i^1, x_i^2) - \alpha d_i, \tag{6.4}$$

$$X_i^2 = \max(x_i^1, x_i^2) - \alpha d_i, \tag{6.5}$$

$$d_i = |x_i^1 - x_i^2|, \tag{6.6}$$

where, α is a positive parameter.

Non-uniform mutation is also used as the mutation operator. For each individual x in a certain generation, the individual x' in the next generation will be generated as follows.

$$\begin{cases} x_k^{'} = x_k + \triangle(t, UB - x_k), \ if \ random \ variable \ \epsilon \ is \ 0, \\ x_k^{'} = x_k - \triangle(t, x_k - LB), \ if \ random \ variable \ \epsilon \ is \ 1, \end{cases} \tag{6.7}$$

$$\triangle(t, y) = y * (1 - r^{1 - \frac{t}{T}}), \tag{6.8}$$

where, x_k is the kth component of individual x and LB and UB are the lower and upper bounds of x_k. $\triangle(t, y)$ returns a value from $[0, y]$. r is a uniform random value in the range of $[0, 1]$ and T is maximal generation.

6.4 Simulations

6.4.1 Training Phase

In this chapter, we use training data set randomly selected from NSL-KDD for Fuzzy GNP-based class association rule mining, that is, the same as that of last chapter. It contains $2,000$ normal connections and $2,000$ misuse intrusion connections consisting of $1,500$ with *neptune* type and 500 with $smurf$ type.

As the number of rules is too large to build the classifier, the two-phase rule pruning mechanism is implemented to pick up useful rules and to reduce the influence from redundant and irrelevant rules in the rule pool. As a result, it remains only 429 normal rules, 437 misuse intrusion rules with *neptune* type and 134 misuse intrusion rules with $smurf$ type after pruning.

6.4.2 Selection of Parameters

In the proposed approach, there are two sets of parameters to be determined. One is the minimum number of points in one block for determining the crowdness of the block. The other is the parameters of the fitness functions in GA. The simulations for setting the above two sets of parameters are conducted using the validation data, which contains $2,000$ normal connections, $2,000$ misuse intrusion connections, including $1,000$ *neptune* and $1,000$ $smurf$.

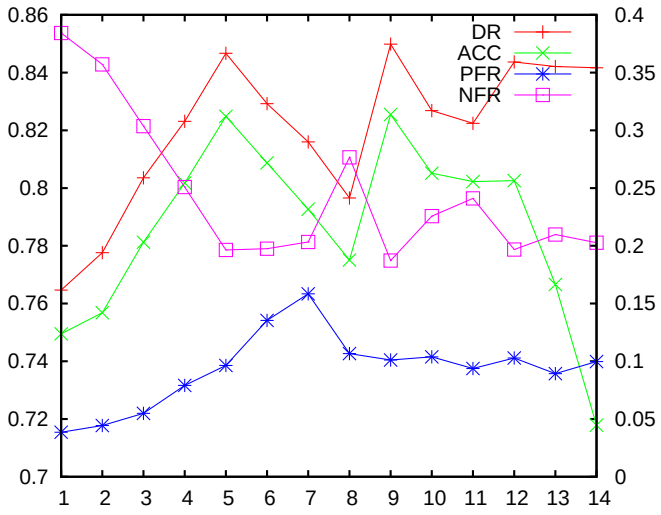


Figure 6.4: The performance with various size of points in the block

Table 6.1: Number of clusters in normal and misuse intrusion class which is obtained by various size of points in the block

setting	C_N	C_I
1	224	92
2	132	92
3	99	92
4	71	92
5	57	92
6	51	92
7	27	92
8	25	92
9	57	76
10	57	67
11	57	59
12	57	55
13	57	36
14	57	18

First, the effects of different size of points in the cluster are analyzed. For this purpose, each parameter of the fitness function is set at 1. Fig. 6.4 shows DR and *Accuracy*, PFR and NFR on different size of points, while Table 6.1 shows the corresponding number of clusters in the normal and misuse class, which was used for calculating Fig. 6.4. Setting $1-5$ indicate that, as the number of clusters in the normal class decreases, DR, *Accuracy* and NFR are improved although PFR increases. If the number of clusters decreases too much, DR and *Accuracy* also decrease, and NFR is not improved any more, but PFR is still increasing, which is shown at setting $6-8$ in Fig. 6.4. So, the number of clusters in the normal class is fixed at 57 like setting 5 and the effect of the setting in the misuse intrusion class is examined. Setting 9 indicates that DR and *Accuracy* increase, and PFR and NFR decrease. However, as the number of clusters in the misuse intrusion decreases, the performance of *Accuracy* decreases a lot. It is concluded from Fig. 6.4 that an appropriate number of clusters (neither large nor small) can improve the performance of intrusion detection system to some extent.

Next, the effects of the parameters of the fitness function are analyzed under the number of clusters with setting 9. They are also evolved by GA on validation data. Fig. 6.5 shows DR, *Accuracy*, PFR and NFR on different parameter settings, while

Table 6.2 shows the corresponding weight settings. When setting β (PFR) and γ (NFR) at small values, DR, *Accuracy* and NFR are worse than those in the case of setting 3 although PFR is a bit better. As α (Accuracy) increases, the performance does not become better, even worse. So, γ is increased after α is set at 1. Then, DR, *Accuracy* and PFR are deteriorated in setting $7-8$. Thus, γ is set at 4. Then, increasing β brings the increase of DR and Accuracy and decrease of PFR and NFR. But, DR, *Accuracy* and NFR become worse by increasing β too much, though PFR has no big change. Therefore, setting 9 is selected.

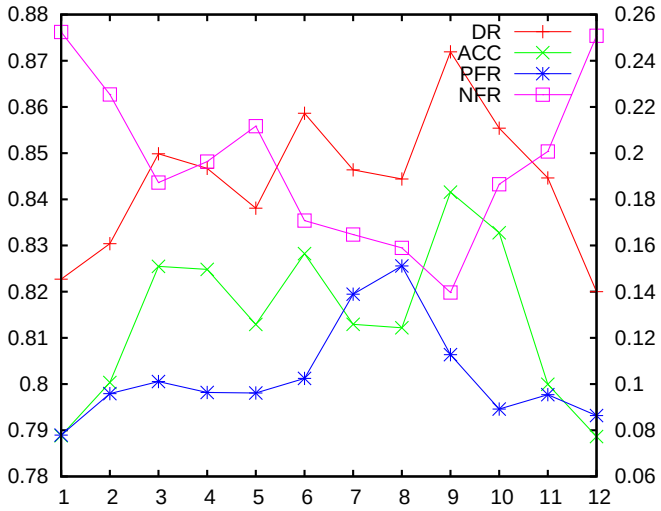


Figure 6.5: Effects of fitness parameter settings

6.4.3 Classification using Gaussian Functions

After determining the shape of Gaussian functions, it can projected on the average matching degree space. Fig. 6.6 shows an example of the shapes of Gaussian functions for three clusters. Two of them belong to the misuse intrusion class like A and B in Fig. 6.6 and one is the normal class like C in Fig. 6.6. As Fig. 6.6 shows, the shapes are ellipse or circle and the boundaries of the classes are different with each other.

Table 6.3 shows the classification results using Gaussian functions using setting 9 in Table 6.1 and Table 6.2. In Table 6.3, $Normal(C)$, $Misuse(C)$ and $Anomaly(C)$ indicate the number of normal, misuse intrusions and anomaly intrusions labeled by the classification with Gaussian functions, respectively, while $Normal(A)$, $Misuse(A)$ and

Table 6.2: Fitness parameter settings

setting	α	β	γ
1	1	0.5	1
2	1	0.5	0.5
3	1	1	1
4	2	1	1
5	5	1	1
6	1	1	4
7	1	1	5
8	1	1	7
9	1	2	4
10	1	5	4
11	1	7	4
12	1	8	4

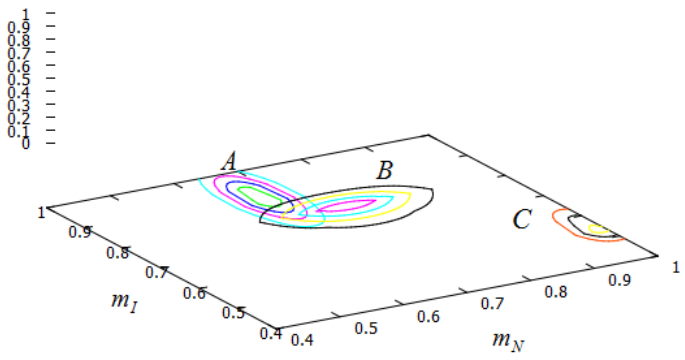


Figure 6.6: An example of the shapes of Gaussian functions in average matching degree space

$Anomaly(A)$ indicate the actual number of normal, misuse intrusions and anomaly intrusions, respectively. It is obvious that most of the anomaly intrusion data can be detected by the proposed classifier, which means it has better detection ability on anomaly intrusions.

According to Table 6.3, DR, $Accuracy$, PFR and NFR are calculated as shown in Eq.(6.9)∼(6.12).

$$DR = (8,616 + 5,036 + 5,320 + 402 + 283)/22,544 = 87.19\% \tag{6.9}$$

Table 6.3: Results of classification with Gaussian functions

	$Normal(C)$	$Misuse(C)$	$Anomaly(C)$	$Total$
$Normal(A)$	$8,616$	61	$1,034$	$9,711$
$Misuse(A)$	3	$5,036$	283	$5,322$
$Anomaly(A)$	$1,789$	402	$5,320$	$7,511$
$Total$	$10,408$	$5,499$	$6,637$	$22,544$

$$Accuracy = (8,616 + 5,036 + 5,320)/22,544 = 84.16\% \tag{6.10}$$

$$PFR = (61 + 1,034)/9,711 = 11.28\% \tag{6.11}$$

$$NFR = (3 + 1,789)/(5,322 + 7,511) = 13.96\% \tag{6.12}$$

Over 70% of anomaly intrusion connection data has been detected. Specially, the number of anomaly intrusion connection data misclassified as misuse intrusion is very small. At the same time, the number of normal connection data misclassified is acceptable.

6.4.4 Comparison with Other Approaches

For comparison, the same NSL-KDD data set is used to do the simulations for comparing the proposed method with other approaches.

One is of the most commonly used neural networks for classification. The architecture used for MLP consists of four layers, i.e., two inputs, two hidden layers(five nodes in each layer) and one output. The average matching degrees $m_N(d)$ and $m_I(d)$ over all training data are also used as the inputs to train the MLP classifier. Fig. 6.7 and Fig. 6.8 show the comparisons of the proposed method with the Distance-based classification, Hybrid classifier, SVM and MLP in terms of DR, $Accuracy$, PFR and NFR.

From the aforementioned comparisons, the performance is improved by the proposed classifier due to the improved detection ability of anomaly intrusion. The proposed method can distinguish normal and anomaly intrusion better than other algorithms. At the same time, it is also effective to detect the misuse intrusion.

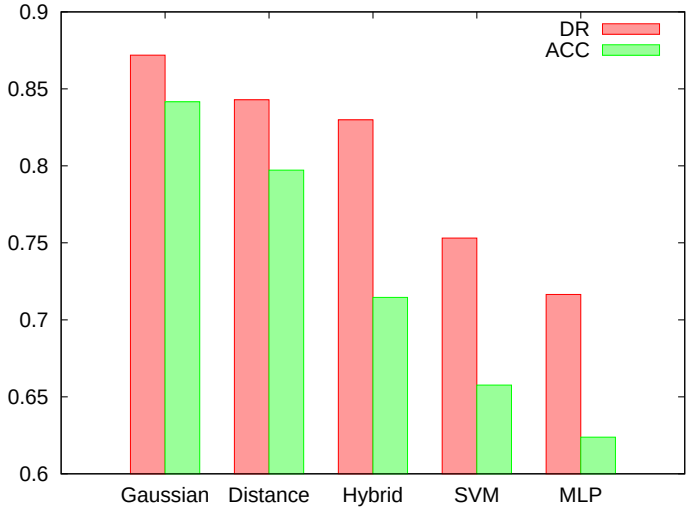


Figure 6.7: DR and ACC comparisons of the proposed method with other algorithms

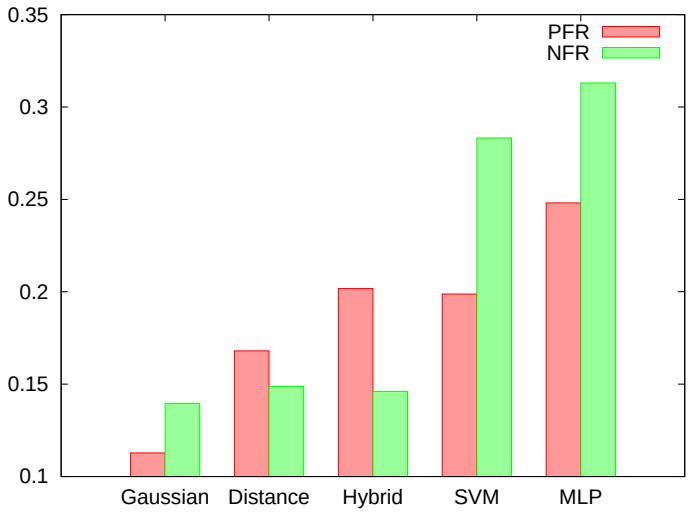


Figure 6.8: PFR and NFR comparisons of the proposed method with other algorithms

6.5 Conclusions

In this chapter, a novel approach is proposed to classify network connections. By distributing the clusters with Gaussian functions, the proposed method can classify a new connection data as normal, misuse intrusion or anomaly intrusion fairly correctly. Most importantly, the proposed approach gathers the similar behaviors as a cluster and builds the Gaussian function for each cluster, where the clustering provides essential

parameters of Gaussian functions and the shapes of Gaussian function are used as the boundary of the cluster. Furthermore, GA is used to decide the shapes of Gaussian functions. The simulation results conducted on NSL-KDD data set show that the detection performance is improved by the proposed method. The comparisons with other approaches indicate that the proposed approach has excellent DR and $Acuuracy$. Especially, the proposed method can distinguish normal and anomaly intrusion correctly comparing with Distance-based approach.

7

Conclusions

In this thesis, a series of data mining methods is proposed for building an efficient intrusion detection system. Intrusion detection systems are analyzed from three aspects: class association rule mining, class association rule pruning and classification.

In chapter 2, a hybrid framework of intrusion detection systems has been proposed to combine the advantages of both misuse detection and anomaly detection. In this system, GNP undertakes extracting class association rules. And it can extract many interesting and important class association rules from network connection data efficiently by using the sub-attributes based on information gain. Information gain can avoid the information loss as much as possible, which can deal with the partition of continuous attributes. These rules are evaluated by the average matching degree of the data with rules. In this way, multi-dimensional data space is converted into a two-dimensional space. A classifier combined misuse detection and anomaly detection has been also proposed by utilizing the average matching degree to classify the new data. By combing misuse detection and anomaly detection, the results show the proposed system has better performance.

However, two problems came out. One is that too many rules brought much useless information into the rule pool and waste much processing time during classification. In this case, a pruning method is needed to reduce the useless rules from the rule pool. Second is that information gain avoided the loss of much information, but the discretization of the continuous attributes into intervals leaded to sharp boundary problem. Therefore, in chapter 3 and 4, we focused on solving the two problems, respectively.

Chapter 3 proposed an efficient class association rule pruning method. This rule pruning method has two stages. The first stage can pre-prune the rules to improve the efficiency of the second step. The second step implemented GA to pick up a small set of effective rules among remaining rules in the first stage. The simulation results show that the proposed two-stage rule pruning method can improve the detection ability of intrusion detection system under a small set of class association rules.

Chapter 4 proposed Fuzzy GNP to extract class association rule. Comparing with conventional GNP, Fuzzy GNP can deal with both discrete and continuous attributes in intrusion detection and overtake the sharp boundary problem of sub-attributes method. Each continuous attribute had its own initial fuzzy membership function and its parameters were evolved along with the GNP evolution. In addition, probabilistic node transition has taken place in the traditional node transition in GNP, which can contribute to extracting diversified rules. The simulation results on KDD Cup 1999 show good detection ability. And Fuzzy GNP can extract more useful rules than conventional GNP.

Chapter 5 and chapter 6 analyzed the classifier for an effective intrusion detection system.

Chapter 5 proposed a Distance-based classification method. Firstly, K-closest neighbor classifier was employed to categorize each new data into either normal or misuse intrusion. Then, the centroids of anomaly intrusion data were defined by the centroids of normal data and misuse intrusion data. Distance-based classification method can distinguish anomaly intrusions from the mixture of normal and misuse intrusions more accurately than other methods. The results on NSL-KDD data set indicate that the detection ability of anomaly intrusions has been improved. From the results on KDD Cup 1999 data set, it is remarkable that the Distance-based classifier can detect all the connection data of *neptune* type and $smurf$ type with known patterns. And the detection performance of this classifier is not sensitive to parameter K.

Chapter 6 proposed a new classifier using Gaussian functions and clustering method. To make full use of normal and misuse intrusion patterns, the proposed method has grouped the similar patterns into the same cluster. Then Gaussian function has been used to look for the boundary for each cluster. GA was used to decide the shapes of Gaussian functions. This classifier can classify the new connection data as normal, misuse intrusion or anomaly intrusion fairly correctly. The proposed method has high

detection performance. Especially, it can distinguish normal and anomaly intrusion well.

Appendix A

Genetic Network Programming(GNP)

A.1 Structure of GNP

As one of evolutionary algorithms(EA)(108)(109), GNP has been proposed as an extension of Genetic Algorithm (GA)(110) and Genetic Programming (GP)(111) in terms of gene structures. GNP uses directed-graphs as genes, whereas GA uses binary strings and GP uses trees. The original motivation for developing GNP is based on the more general representation ability of graphs compared with that of strings in GA or trees in GP in dynamic environment.(112)(113)(114)

As Fig. A.1 shows, one GNP individual is composed of one start node, plural judgment nodes and processing nodes. Start node has no function and no conditional branch. The only role of the start node is to determine the first node to be executed. Judgment nodes judge the information from the environments and determine what the next node is. Processing nodes describe action/processing functions of GNP. In contrast to judgment nodes, processing nodes have no conditional branch. By separating processing and judgment functions, various combinations of judgment and processing can be handled by GNP. That is, the fitness of different combinations of judgment and processing functions in GNP can be evaluated through the process of evolution.

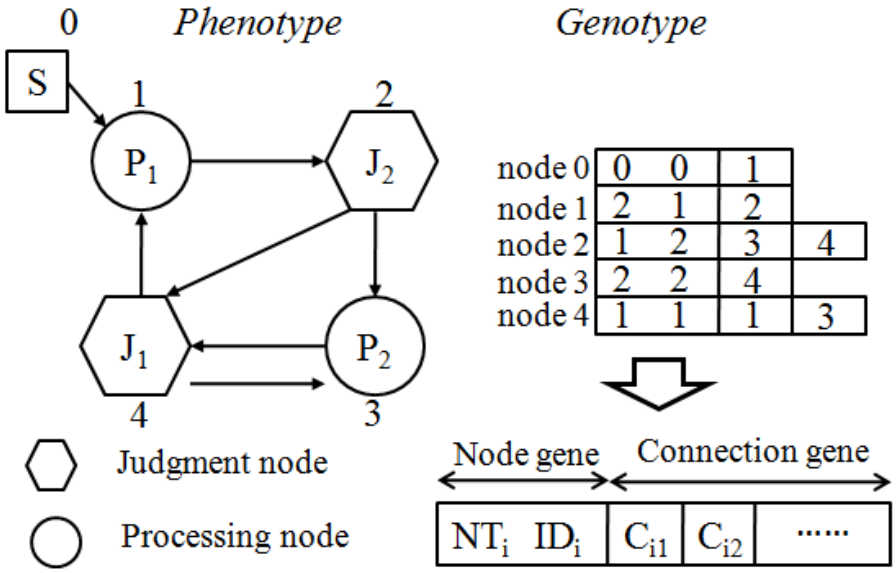


Figure A.1: Basic structure of GNP

A.2 Operators of GNP

GNP are also evolved by performing genetic operators, e.g., selection, crossover and mutation.(112)

1. Selection: The purpose of selection operator is to select individuals according to their fitness. In Fig. A.2, some selection mechanisms are shown.

   (a) Roulette Selection: The probability that individuals are selected is proportional to their fitness values.

   (b) Tournament Selection: First, a subset of population is randomly selected for tournament selection. Then, the winner individual in this subset is chosen as the selected individual.

   (c) Elite Selection: The best individual of population is selected.

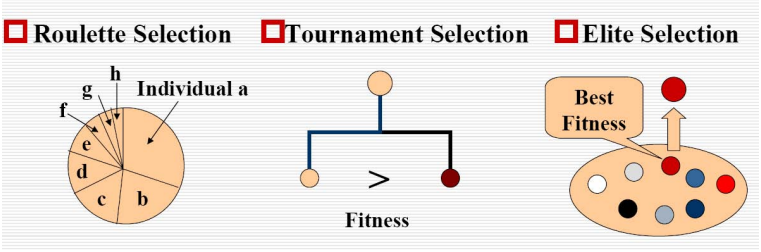


Figure A.2: Selection methods in GNP

2. Crossover: Crossover is executed between two parents and generates two offspring. Two parents individuals are firstly selected using selection mechanism. Then, each node i is selected as a crossover node with the probability of $P_c (0 \leq P_c \leq 1)$. The value of P_c determines the exploration ability of GNP. Thirdly, two parents exchange the genes of the corresponding crossover nodes. Finally, generated new individuals become the new ones of the next generation. In Fig. A.3, a crossover example of GNP is shown. The red parts of parents are exchanged in offspring after crossover operation.

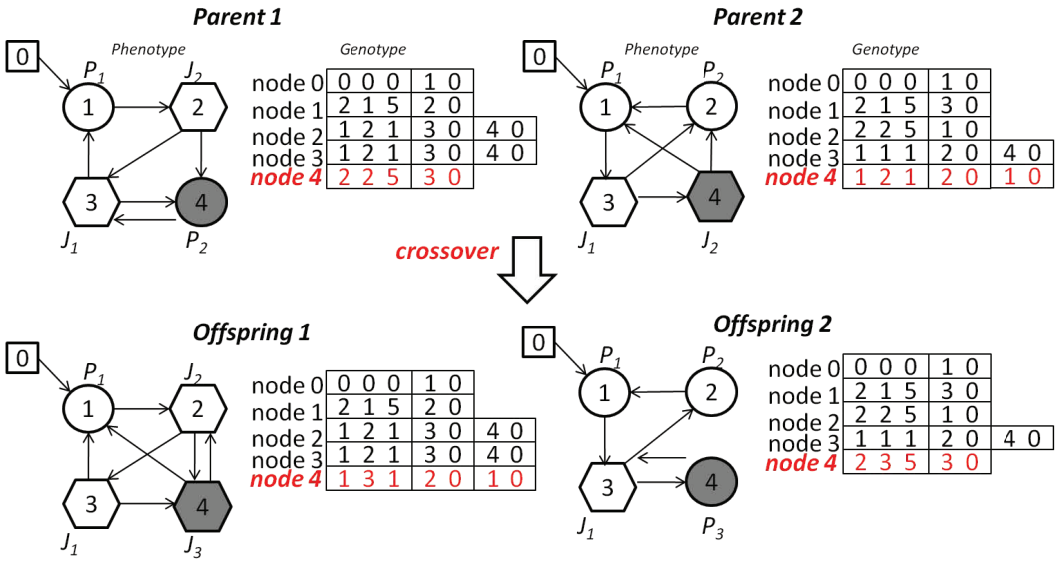


Figure A.3: Crossover in GNP

3. Mutation: Mutation is executed in one individual and a new one is generated.

In detail, one individual is firstly selected as a parent for mutation using selection mechanism. Secondly, each node i is selected as a mutation node with the probability of $P_m (0 \leq P_m \leq 1)$. Finally, the contents of selected nodes and their connections are changed randomly. The purpose of mutation is to find the global optimal solution instead of the local optimal solution. In Fig. A.4, a mutation example of GNP is shown. The red part in the parent is randomly changed to other value.

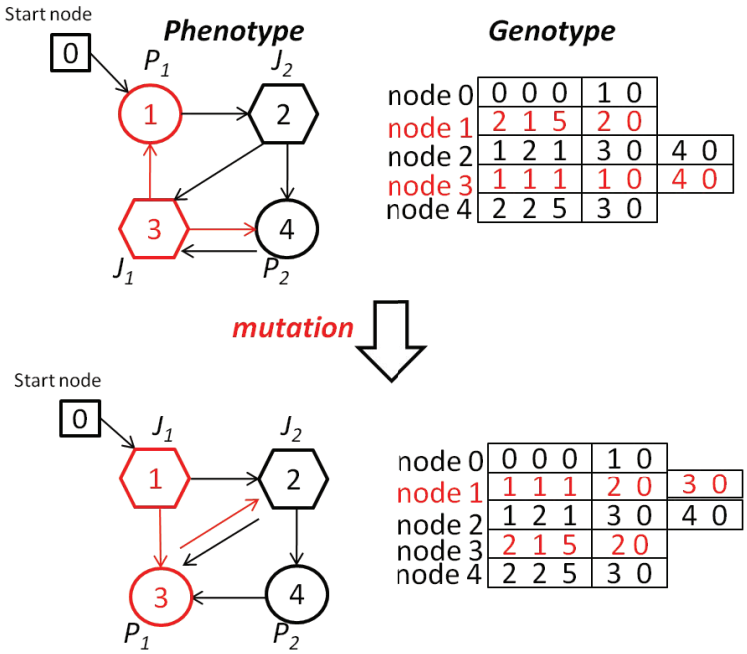


Figure A.4: Mutation in GNP

Appendix B

Class Association Rule Mining

B.1 Association Rule Mining

An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The meaning of this association rule is that if the antecedent X is satisfied, then the consequent Y is also satisfied. It is easy to extract a large number of candidate association rules. How to evaluate the rules is a key point. Therefore, interestingness measures are used to select interesting association rules from the set of candidate rules. For a given rule $X \Rightarrow Y$, various measures are devised using the frequency counts as show in the Table B.1, i.e., Support, Confidence, χ^2 value and etc. In Table B.1, N means the total number of records. n_X is the number of records where X is satisfied, while $n_{\bar{X}}$ means the number of records where X is not satisfied, and the corresponding notions of Y are defined in the same way. The rate of tuples satisfying X in the training database is called the support of X, denoted by $support(X) = n_{XY}/N$. The confidence of rule $X \Rightarrow Y$ is defined as the ratio of $support(X \cup Y)/support(X)$, denoted by $confidence(X \Rightarrow Y) = n_{XY}/n_X$.

Table B.1: The Frequency counts table of X and Y

	Y	$\bar{Y}$	Total
X	n_{XY}	$n_{X\bar{Y}}$	n_X
$\bar{X}$	$n_{\bar{X}Y}$	$n_{\bar{X}\bar{Y}}$	$n_{\bar{X}}$
Total	n_Y	$n_{\bar{Y}}$	N

Assume $support(X) = x$, $support(Y) = y$, $support(X \cup Y) = z$ and the total

number of tuples in training database is N, then the χ^2 value of rule $X \Rightarrow Y$ can be calculated as

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

If the χ^2 value is higher than a predefined threshold, the assumption that X and Y are dependent should be accepted. (3.84 at the 95% significance level or 6.64 at the 99% significance level).

Class Association Rule has the different form with association rule. It has a class label as its consequent part. The following shows an example of a class association rule.

$$(A_m = 1) \wedge \cdots \wedge (A_n = 1) \Rightarrow (k \in C), \tag{B.2}$$

where A_i is an attribute of database with value 1 or 0 (1 means satisfied, 0 means not satisfied), k is the class label and C is the set of suffixes of classes. Class association rule can be viewed as a special case of the association rule $X \Rightarrow Y$ with a fixed consequent.

B.2 Class Association Rule Mining using GNP

In GNP-based Class Association Rule Mining, attributes and their values of rules correspond to the functions of judgment nodes in GNP. And the connection of judgment nodes can represent candidate class association rules. An example of the GNP representation is displayed in Fig. B.1. Processing node P_1 serves as the beginning of class association rules. $A_1 = 1$, $A_2 = 1$ and $A_3 = 1$ denote the functions of judgment nodes J_1, J_2 and J_3 respectively. For example, the candidate class association rules, such as $(A_1 = 1) \Rightarrow (C \in k)$, $(A_1 = 1) \wedge (A_2 = 1) \Rightarrow (C \in k)$ and so on, can be represented by GNP in Fig. B.1. In Fig. B.1, N indicates the number of total tuples in the training database; a , b and c are the numbers of tuples moving to the Yes-side at each judgment node; $a(k)$, $b(k)$ and $c(k)$are the numbers of tuples moving to the Yes-side at each judgment node under the condition of belonging to class k. Assume the number of tuples belonging to class k is $y(k)$, which can be calculated by counting the number of tuples in the training database that have a class label k, then value of χ^2 value, support value and confidence value of classification rule $(A_1) \Rightarrow (C \in k)$ can

be calculated as follows in Eq.(B.3)-(B.5):

$$support((A_1) \Rightarrow (C \in k)) = \frac{a}{N}, \tag{B.3}$$

$$confidence((A_1) \Rightarrow (C \in k)) = \frac{a(k)}{a}, \tag{B.4}$$

$$\chi^2 = \frac{N(a(k) - ay(k)/N)^2}{ay(k)(1 - a/N)(1 - y(k)/N)}. \tag{B.5}$$

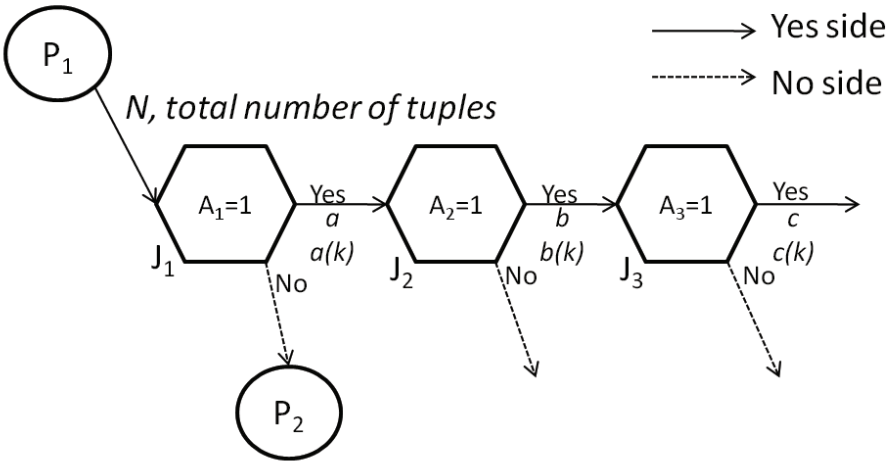


Figure B.1: GNP representation of class association rules

References

[1] F. Cohen. **Computer Viruses: Theory and Experiments**. *In Proc. of the 7th DOD/NBS Computer Security Conference*, pages 240–263, 1984.

[2] M. Bishop. **Computer Security in the Future**. *The ISC International Journal of Information Security*, **3(1)**:3–27, 2011.

[3] SANS. *SANS Institute-Intrusion Detection FAQ http://www.sans.org/resources/idfag/*, 2012.

[4] B. Mukherjee, L. T. Heberlein, and K. N. Levitt. **Network intrusion detection**. *Network, IEEE*, **8(3)**:26–41, 1994.

[5] M. Bishop. **Introduction to Computer Security**. *Addison Wesley Professional*, 2004.

[6] ISC. *The ISC Domain Survey. https://www.isc.org/solutions/survey/*, 2012.

[7] P. Lyman, H. R. Varian, P. Charles, N. Good, L. L. Jordan, J. Pal, and K. Swearingen. **How much Information**. *http://www2.sims.berkeley.edu/research/projects/how-much-info-2003*, 2003.

[8] K. Hwang, Y. Chen, and H. Liu. **Defending Distributed Computing Systems from Malicious Intrusions and Network Anomalies**. *In Proc. of the 19th IEEE International Parallel and Distributed Processing Symposium*, 2005.

[9] K. Hwang, Y. Kwok, S. Song, M. Cai, Y. Chen, and Y. Chen. **DHT-Based Security Infrastructure for Trusted Internet and Grid Computing**. *International Journal Of Critical Infrastructures*, **2(4)**:412–433, 2006.

[10] Z. Yu, J. J. P. Tsai, and T. Weigert. **An Automatically Tuning Intrusion Detection System**. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **37(2)**:373–384, 2007.

[11] A. Sundaram. **An Introduction to Intrusion Detection**. *Special Issue on Computer Security*, **2(4)**:3–7, 1996.

[12] R. A. Kemmerer. **Intrusion Detection: A Brief History and Overview**. *Computer*, **35(4)**:27–30, 2002.

[13] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz. **An Intelligent Intrusion Detection System (IDS) for Anomaly and Misuse detection in Computer Networks**. *Expert Systems with Applications*, **29(4)**:713-722, 2005.

[14] V. Chandola, A. Banerjee, and V. Kumar. **Anomaly Detection: A Survey**. *ACM Computing Surveys (CSUR)*, **41(3)**, 2009.

[15] A. Patcha and J. M. Park. **An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends**. *Computer Networks*, **51(12)**:3448-3470, 2007.

[16] J. Hu, X. Yu, D. Qiu, and H. H. Chen. **A Simple and Efficient Hidden Markov Model Scheme for Host-based Anomaly Intrusion Detection**. *Network, IEEE*, **23(1)**:42–47, 2009.

[17] D. Y. Yeung and Y. Ding. **Host-based Intrusion Detection using Dynamic and Static Behavioral Models**. *Pattern Recognition*, **36(1)**:229-243, 2003.

[18] L. Vokorokos and A. Balaz. **Host-based Intrusion Detection System**. *In Proc. of the 14th International Conference on Intelligent Engineering Systems (INES)*, pages 43–47, 2010.

[19] D. Mutz, F. Valeur, G. Vigna, and C. Kruegel. **Anomalous System Call Detection**. *ACM Transactions on Information and System Security (TISSEC)*, **9(1)**:61–93, 2006.

[20] W. Hu, W. Hu, and S. Maybank. **AdaBoost-Based Algorithm for Network Intrusion Detection**. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **38(2)**:577–583, 2008.

[21] S. Chebrolu, A. Abraham, and J. P. Thomas. **Feature Deduction and Ensemble Design of Intrusion Detection Systems**. *Computer and Security*, **24(4)**:295–307, 2005.

[22] D. Wagner and P. Soto. **Mimicry Attacks on Host-based Intrusion Detection Systems**. *In Proc. of the 9th ACM conference on Computer and Communications Security*, pages 255–264, 2002.

[23] G. Vigna and R. A. Kemmerer. **NetSTAT: A Network-based Intrusion Detection Approach**. *In Proc. of the 14th Annual Computer Security Applications Conference*, pages 25–34, 1998.

[24] J. Zhang, M. Zulkernine, and A. Haque. **Random-Forests-Based Network Intrusion Detection Systems**. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, **38(5)**:649–659, 2008.

[25] H. Han, X. Lu, and L. Ren. **Using Data Mining to Discover Signatures in Network-based Intrusion Detection**. *In Proc. of the International Conference on Machine Learning and Cybernetics*, **1**:13–17, 2002.

[26] M. Lehtinen and A. C. Lear. **Intrusion Detection: Managing the Risk of Connectivity**. *IT Professional*, **1(6)**:11–13, 1999.

[27] C. Herringshaw. **Detecting Attacks on Networks**. *Computer*, **30(12)**:16–17, 1997.

[28] D. E. Denning. **An Intrusion Detection Model**. *IEEE Trans. on Software Engineering*, **SE-13(2)**:222–232, 1987.

[29] N. Ye, X. Li, Q. Chen, S. M. Emran, and M. Xu. **Probabilistic Techniques for Intrusion Detection based on Computer Audit Data**. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, **31(4)**:266–274, 2001.

[30] S. Mabu, C. Chen, N. Lu, K. Shimada, and K. 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*, **41(1)**:130–139, 2011.

[31] K. K. Gupta, B. Nath, and R. Kotagiri. **Layered Approach Using Conditional Random Fields for Intrusion Detectionn**. *IEEE Transactions on Dependable and Secure Computing*, **7(1)**:35–49, 2010.

[32] J. P. Anderson. **Computer Security Threat Monitoring and Surveillance**. *Technical report, James P. Anderson Company, Fort Washington, Pennsylvania*, 1980.

[33] S. Kumar and E. H. Spafford. **Pattern Matching Model for Misuse Intrusion Detection**. *In Proc. of the 17th National Computer Security Conference*, pages 11–21, 1994.

[34] C. Zhou, Y. Liu, and H. Zhang. **A Pattern Matching based Network Intrusion Detection System**. *In Proc. of the 9th International Conference on Control, Automation, Robotics and Vision (ICARCV 06)*, pages 1–4, 2006.

[35] J. Li and C. Manikopoulos. **Early Statistical Anomaly Intrusion Detection of DOS Attacks using MIB Traffic Parameters**. *Information Assurance Workshop, IEEE Systems, Man and Cybernetics Society*, page 53-59, 2003.

[36] W. Teng, M. Hsieh, and M. Chen. **A Statistical Framework for Mining Substitution Rules**. *Knowledge and Information Systems*, **7(2)**:158-178, 2005.

[37] W. Lee and D. Xiang. **Information-Theoretic Measures for Anomaly Detection**. *In Proc. of the 2001 IEEE Symposium on Security and Privacy*, pages 130–143, 2001.

[38] W. Lee, S. Stolfo, P. K. Chan, E. Eskin, W. Fan, M. Miller, S. Hershkop, and J. Zhang. **Real Time Data Mining-based Intrusion Detection**. *In DARPA Information Survivability Conference and Exposition II*, pages 85–100, 2001.

[39] S. T. Brugger. **Data Mining Methods for Network Intrusion Detection**. *UC Davis Dissertation Proposal*, 2004.

[40] P. D. Williams, K. P. Anchor, J. L. Bebo, G. H. Gunsch, and G. D. Lamont. **CDIS: Towards a Computer Immune System for Detecting Network Intrusions**. *In Proc. of the 4th International Symposium on Recent Advances in Intrusion Detection*, pages 117–133, 2001.

[41] P. Laskov, P. Dssel, C. Schfer, and K. Rieck. **Learning Intrusion Detection: Supervised or Unsupervised?** *Image Analysis and ProcessingICIAP 2005 Lecture Notes in Computer Science*, **3617/2005**:50–57, 2005.

[42] R. Sommer and V. Paxson. **Outside the Closed World: on using Machine Learning for Network Intrusion Detection**. *2010 IEEE Symposium on Security and Privacy (SP)*, pages 305–316, 2010.

[43] C. Sinclair, L. Pierce, and S. Matzner. **An Application of Machine Learning to Network Intrusion Detection**. *In Proc. of the 15th Annual Computer Security Applications Conference(ACSAC '99)*, pages 371–377, 1999.

[44] N. Ye, S. M. Emran, X. Li, and Q. Chen. **Statistical Process Control for Computer Intrusion Detection**. *In Proc. of DARPA Information Survivability Conference and Exposition II(DISCEX'01)*, **1**:3–14, 2001.

[45] A. Qayyum, M. H. Islam, and M. Jamil. **Taxonomy of Statistical based Anomaly Detection Techniques for Intrusion Detection**. *In Proc. of the IEEE Symposium on Emerging Technologies*, pages 270–276, 2005.

[46] T. Verwoerd and R. Hunt. **Intrusion Detection Techniques and Approaches**. *Computer Communications*, **25(15)**:1356–1365, 2002.

[47] D. Anderson, T. F. Lunt, H. Javits, A. Tamaru, and A. Valdes. **Detecting Unusual Program Behavior using the Statistics Components of NIDES**. *NIDES Technical Report, SRI International*, 1995.

[48] J. B. D. Caberera, B. Ravichandran, and R. K. Mehra. **Statistical Traffic Modeling for Network Intrusion Detection**. *In Proc. of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 466–473, 2000.

[49] N. Ye. **A Markov Chain Model of Temporal Behavior for Anomaly Detection**. *In Proc. of the IEEE SMC Inform. Assurance Security Workshop*, pages 166–169, 2000.

[50] A. Kanaoka and E. Okamoto. **Multivariate Statistical Analysis of Network Traffic for Intrusion Detection**. *In Proc. of the 14th International Workshop onDatabase and Expert Systems Applications*, pages 472–476, 2003.

[51] N. B. Amor, S. Benferhat, and Z. Elouedi. **Naive Bayes vs Decision Tree in Intrusion Detection**. *In Proc. of the 2004 ACM symposium on Applied computing*, 2004.

[52] S. Mukkamala, G. Janoski, and A. Sung. **Intrusion Detection using Neural Networks and Support Vector Machines**. *In Proc. of the Int. Joint Conf. Neural Netw.*, **2**:1702–1707, 2002.

[53] A. Rapaka, A. Novokhodko, and D. Wunsch. **Intrusion Detection using Radial Basis Function Network on Sequences of System Calls**. *In Proc. of Int. Joint Conf. Neural Netw.*, **3**:1820–1825, 2003.

[54] F. Karray and C. Silva. **Soft Computing and Intelligent Systems Design: Theory, Tools and Applications**. *Addison Wesley Publishing*, 2004.

[55] J. Hertz, A. Krogh, and R. Palmer. **Introduction to the Theory of Neural Computation**. *Addison Wesley Publishing*, 1991.

[56] J. Cannady. **Artificial Neural Networks for Misuse Detection**. *In Proc. Of the 1998 National Information Systems Security Conference*, pages 443–456, 1998.

[57] D. Parikh and T. Chen. **Data Fusion and Cost Minimization for Intrusion Detection**. *IEEE Transactions on Information Forensics and Security*, **3(3)**:381–389, 2008.

[58] Y. Freund and R. E. Schapire. **A Decision-Theoretic Generalization of On-line Learning and An Application to Boosting**. *Lecture Notes in Computer Science*, **904/1995**:23–37, 1995.

[59] L. Portnoy, E. Eskin, and S. Stolfo. **Intrusion Detection with Unlabeled Data using Clustering**. *In Proc of the ACM Workshop Data Mining Applied to Security(DMSA)*, 2001.

[60] H. Shah, J. Undercoffer, and A. Joshi. **Fuzzy Clustering for Intrusion Detection**. *In Proc. of the 12th IEEE Int'l Conf. Fuzzy Systems(FUZZY-IEEE'03)*, **2**:1274–1278, 2003.

[61] A. J. Hoglund, K. Hatonen, and A. S. Sorvari. **A Computer Host-based User Anomaly Detection System using the Self-Organizing Map**. *In Proc. of Int. Joint Conf. Neural Netw.*, **5**:411–416, 2000.

[62] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood. **On the Capability of An SOM based Intrusion Detection System**. *In Proc. of Int. Joint Conf. Neural Netw.*, **3**:1808–1813, 2003.

[63] W. Lee and S. J. Stolfo. **A Framework for Constructing Features and Models for Intrusion Detection Systems**. *ACM Trans. on Information and System Security*, **3(4)**:227–261, 2000.

[64] W. Lee, S. J. Stolfo, and K. W. Mok. **Mining Audit Data to Build Intrusion Detection Models**. *In Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 66–72, 1998.

[65] L. A. Zadeh. **Fuzzy Logic, Neural Networks, and Soft Somputing**. *Communications of the ACM*, **37(3)**:77–84, 1994.

[66] J. E. Dickerson, J. Juslin, O. Koukousoula, and J. A. Dickerson. **Fuzzy Intrusion Detection**. *IFSA World Congress and 20th NAFIPS International Conference*, **3**:1506–1510, 2001.

[67] A. Tajbakhsh, M. Rahmati, and A. Mirzaei. **Intrusion Detection using Fuzzy Association Rules**. *Applied Soft Computing*, **9(2)**:462–469, 2009.

[68] D. E. Goldberg. **Genetic Algorithm in Search, Optimization and Machine Learning**. *Addison-Wesley*, 1989.

[69] K. Shazzad and J. Park. **Optimization of Intrusion Detection through Fast Hybrid Feature Selection**. *In Proc. of the Sixth International Conference on Parallel and Distributed Computing Applications and Technologies*, page 264-267, 2005.

[70] A. Hofmann, T. Horeis, and B. Sick. **Feature Selection for Intrusion Detection: An Evolutionary Wrapper Approach**. *In Proc. of the 2004 IEEE International Joint Conference on Neural Networks*, **2**:1563–1568, 2004.

[71] D. Kim, H. Nguyen, and J. Park. **Genetic Algorithm to Improve SVM Based Network Intrusion Detection System**. *In Proc. of the 19th International Conference on Advanced Information Networking and Applications*, page 155-158, 2005.

[72] T. Hong, C. Chen, Y. Lee, and Y. Wu. **Genetic-Fuzzy Data Mining with Divide-and-Conquer Strategy**. *IEEE Trans. on Evolutionary Computation*, **12(2)**:252–265, 2008.

[73] M. J. Zaki. **Generating Non-redundant Association Rules**. *In Proc. of the sixth ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pages 34–43, 2000.

[74] M. Kryszkiewicz. **Representative Association Rules and Minimum Condition Maximum Consequence Association Rules**. *Lecture Notes in Computer Science*, **1510/1998**:361–369, 1998.

[75] J. Li, H. Shen, and R. Topor. **Mining the Optimal Class Association Rule Set**. *Knowledge-based Systems*, **15(7)**:399–405, 2002.

[76] B. Lent, A. Swami, and J. Widom. **Clustering Association Rules**. *In proc. of the 13th International Conference on Data Engineering*, pages 220–231, 1997.

[77] G. Folino, C. Pizzuti, and G. Spezzano. **GP Ensemble for Distributed Intrusion Detection Systems**. *Pattern Recognition and Data Mining Lecture Notes in Computer Science*, **3686/2005**:54–62, 2005.

[78] 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 on Evolutionary Computation*, pages 1276–1282, 2001.

[79] N. Lu, S. Mabu, and K. Hirasawa. **Integrated Rule Mining based on Fuzzy GNP and Probabilistic Classification for Intrusion Detection**. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **15(5)**, 2011.

[80] G. B. White, E. A. Fisch, and U. W. Pooch. **Cooperating Security Managers: A Peer-Based Intrusion Detection System**. *Network, IEEE*, **10(1)**:20–23, 1996.

[81] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. **A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data**. *Applications of Data Mining in Computer Security*, 2002.

[82] W. Fan, M. Miller, S. Stolfo, W. Lee, and P. Chan. **Using Artificial Anomalies to Detect Unknown and Known Network Intrusions**. *Knowledge and Information Systems*, **6(5)**:507–527, 2004.

[83] K. S. Killourhy and R. A. Maxion. **Undermining an Anomaly-based Intrusion Detection System Using Common Exploits**. *In Proc. of the 5th International Conference on Recent Advances in Intrusion Detection*, pages 54–73, 2002.

[84] D. Barbar, J. Couto, S. Jajodia, L. Popyack, and N. Wu. **ADAM: Detecting Intrusions by Data Mining**. *In Proc. of the IEEE Workshop on Information Assurance and Security*, pages 11–16, 2001.

[85] KDD Cup 1999. **Intrusion Detection Data**. *http://kdd.ics.uci.edu/databases/kddcup99*.

[86] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan. **Cost based Modeling for Fraud and Intrusion Detection: Results from the Jam Project**. *In Proc. of DARPA Information Survivability Conference and Exposition*, **2**:130–144, 2000.

[87] K. Shimada, K. Hirasawa, and J. Hu. **Class Association Rule Mining with Chi-Squared Test Using Genetic Network Programming**. *IEEE International Conference on Systems, Man and Cybernetics*, **6**:5338–5344, 2006.

[88] B. Liu, W. Hsu, and Y. Ma. **Integrating Classification and Association Rule Mining**. *In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 80–86, 1998.

[89] H. Ishibuchi, T. Nakashima, and T. Morisawa. **Voting in Fuzzy Rule-based Systems for Pattern Classification Problems**. *Fuzzy Sets and Systems*, **103(2)**:223–238, 1999.

[90] H. Ishibuchi and T. Yamamotoa. **Rule Weight Specification in Fuzzy Rule-Based Classification Systems**. *IEEE Trans. on Fuzzy Systems*, **13(4)**:428–435, 2005.

[91] L. Geng. **Interestingness Measures for Data Mining: A Survey**. *ACM Computing Surveys*, **383**:1–32, 2006.

[92] H. Toivonen, M. Klemettinen, P. Ronkainen, and H. Manilla. **Pruning and Grouping Discovered Association Rules**. *In Workshop on Statistics, Machine Learning and Knowledge Discovery in Databases*, pages 47–52, 1995.

[93] P. N. Tan, V. Kumar, and J. Srivastava. **Selecting the Right Interestingness Measure for Association Patterns**. *In Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 32–41, 2002.

[94] E. Gonzales, Shingo S. Mabu, K. Taboada, K. Shimada, and K. Hirasawa. **Efficient Pruning of Class Association Rules Using Statistics and Genetic Relation Algorithm**. *SICE Journal of Control, Measurement, and System Integration*, **3(5)**:336–345, 2011.

[95] Z. Michalewicz. **Genetic Algorithms + Data Structures = Evolution Programs**. *Springer-Verlag Berlin Heidelberg, New York*, 1996.

[96] D. Song, M. I. Heywood, and A. N. Zicir-Heywood. **A Linear Genetic Programming Approach to Intrusion Detection**. *GECCO 2003 Lecture Notes in Computer Science*, **2724**:2325–2336, 2003.

[97] Z. Bankovic, D. Stepanovic, S. Bojanic, and O. N. Taladriz. **Improving Network Security using Genetic Algorithm Approach**. *Journal of Computers and Electrical Engineering*, **335-6**:438–451, 2007.

[98] L. A. Zadeh. **Fuzzy Sets**. *Information and Control*, **8(3)**:338–353, 1965.

[99] A. Kandel. **Fuzzy Expert Systems**. *Boca Raton, FL: CRC*, 1992.

[100] K. Taboada, S. Mabu, E. Gonzales, K. Shimada, and K.Hirasawa. **Mining Fuzzy Association Rules: A General Model Based on Genetic Network Programming and its Applications**. *IEEJ Transactions on Electrical and Electronic Engineering*, **5(3)**:343-354, 2010.

[101] S. M. Bridges and R. B. Vaughn. **Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection**. *In Proc. of the National Information Systems Security Conference (NISSC)*, 2000.

[102] J. Han and J. Pei. **CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules**. *In Proc. of IEEE International Conference on Data Mining*, pages 369–376, 2001.

[103] T. Cover and P. Hart. **Nearest Neighbor Pattern Classification**. *IEEE Transactions on Information Theory*, **13(1)**:21–27, 1967.

[104] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. **A Detailed Analysis of the KDD CUP 99 Data Set**. *In Proc. of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Application*, pages 1–6, 2009.

[105] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. **NSLKDD Intrusion Detection Data**. *http://nsl.cs.unb.ca/NSL-KDD/*.

[106] C. Chang and C. Lin. **LIBSVM-A Library for Support Vector Machines**. *http://www.csie.edu.tw/ cjlin/libsvm/*.

[107] N. Lu, S. Mabu, T. Wang, and K. Hirasawa. **Distance-based Classification using Average Matching Degree and its Application to Intrusion Detection Systems**. *IEEJ Transactions on Electronics, Information and Systems*, **132(12)**, 2012.

[108] D. B. Fogel. **An Introduction to Simulated Evolutionary Optimization**. *IEEE Transactions on Neural Networks*, **5(1)**:3–14, 1994.

[109] L. J. Fogel, A. J. Owens, and M. J. Walsh. **Artificial Intelligence through Simulated Evolution**. *John Wiley and Sons*, 1966.

[110] S. N. Sivanandam and S. N. Deepa. **Introduction to Genetic Algorithms**. *Springer Publishing Company*, 2007.

[111] J. Koza and R. Poli. **Genetic Programming**. *Search Methodologies*, pages 127–164, 2005.

[112] 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 Journals*, **15(3)**:369–398, 2007.

[113] K. Hirasawa, T. Eguchi, J. Zhou, L. Yu, J. Hu, and S. Markon. **A Double-Deck Elevator Group Supervisory Control System Using Genetic Network Programming**. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, **38(4)**:535–550, 2008.

[114] T. Eguchi, K. Hirasawa, J. Hu, and N. Ota. **A Study of Evolutionary Multiagent Models based on Symbiosis**. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **36(1)**:179–193, 2006.

List of Publications

Journals

J1 Nannan Lu, Shingo Mabu, Tuo Wang and Kotaro Hirasawa, "Distance-based Classification using Average Matching Degree and its Application to Intrusion Detection Systems", *IEEJ Transactions on Electronics, Information and Systems*, Vol. 132, No. 12, accepted. (2012/08/14)

J2 Nannan Lu, Shingo Mabu, Tuo Wang and Kotaro Hirasawa, "An Efficient Class Association Rule Pruning Method for Unified Intrusion Detection System using Genetic Algorithm", *IEEJ Transactions on Electrical and Electronic Engineering*, Vol. 8, No. 2, accepted. (2012/01/16)

J3 Nannan Lu, Shingo Mabu and Kotaro Hirasawa, "Integrated Rule Mining based on Fuzzy GNP and Probabilistic Classification for Intrusion Detection", *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 15, No. 5, pp. 495-505, 2011.

International Conferences (with Review Process)

C1 Nannan Lu, Shingo Mabu, Tuo Wang and Kotaro Hirasawa, "Integrated Fuzzy GNP Rule Mining with Distance-based Classification for Intrusion Detection System", *In Proc. of the IEEE International Conference on systems, Man and Cybernetics*, 2012. 10, accepted.

C2 Nannan Lu, Shingo Mabu, Tuo Wang and Kotaro Hirasawa, "Efficient Hybrid Rule Pruning for Intrusion Detection using Multi-dimensional Probability Distribution", *In Proc. of the SICE International Annual Conference*, pp. 2822-2828, 2011.

C3 Nannan Lu, Shingo Mabu, Wenjng Li and Kotaro Hirasawa, "Hybrid Rule Mining based on Fuzzy GNP and Probabilistic Classification for Intrusion Detection", *In Proc. of the SICE International Annual Conference*, pp. 2614-2619, 2010.

C4 Shingo Mabu, Wenjng Li, Nannan Lu, Yu Wang and Kotaro Hirasawa, "Classification based on a multi-dimensional probability distribution and its application to network intrusion detection", *In Proc. of the 2010 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-7, October, 2010.