Network Intrusion Detection
By
Artificial Immune System

A thesis submitted in fulfilment of
requirements for the degree of
MASTER OF ENGINEERING

Junyuan Shen

ELECTRICAL AND COMPUTER ENGINEERING
COLLEGE OF SCIENCE, ENGINEERING AND HEALTH
RMIT UNIVERSITY

MARCH 2012

Dedicated to my parents, and my supervisor

Preface

Abstract

With computer network's fast penetration into our life, various types of malicious attacks and service abuses increase dramatically. Network security has become one of the big challenges in the modern networks. Intrusion Detection (ID) is one of the active branches in network security research field. Many technologies, such as neural networks, fuzzy logic and genetic algorithms have been applied in intrusion detection and the results are varied. In this thesis, an Artificial Immune System (AIS) based intrusion detection is explored. AIS is a bio-inspired computing paradigm that has been applied in many different areas including intrusion detection. The main objective of our research is to improve the AIS based Intrusion Detection System's (IDS) performance on detection while keeping its system computing complexity to a low level.

An IDS requires specified monitoring parameter set. In a computer network, there are many parameters can be collected or monitored. The quantity of parameters could be real big. These parameters can be used for the intrusion detection purpose. However, the significance of these parameters in intrusion detection can be very different. If all parameters were used, the computing complexity of IDS would be high. Therefore the selection of a group of significant parameters is necessary. This process is called feature selection. Two feature selection algorithms, i.e. Rough set algorithm (RSA) and linear genetic programming (LPG) are selected and compared in this thesis. An improved AIS based IDS with these two feature selection algorithms are studied.

A basic feature selection algorithm only picks the features to be used, assuming they have equal contribution towards the system performance and that is not the case in reality. Therefore weighing the parameters' contribution in the IDS is expected to further improve the performance. However, assigning weights to the selected features is not an easy work. In this thesis, a weight distribution scheme among selected features is proposed. With a simplified exhausted approach, an optimal weight allocation is obtained. The results show that the improved AIS based IDS with

weighted feature selection can achieve 99.98 % of true positive rate while keeping the true negative rate at 99.94%. These results are obtained from the experiment on the popular testing dataset: KDD Cup 99. The results indicate the proposed scheme outperforms most of the existing IDS on the same testing data set.

Declaration

I certify that the work presented in this thesis, except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

Signature: _____

Junyuan Shen

Date: 1/28/2012

Acknowledgement

This research would not have been possible without the support of my supervisor, Dr Jidong Wang from the School of Electrical and Computer Engineering. I would like to express my sincere gratitude for his patient guidance, suggestions and support during the progress of the research. This also extends to all the staff of the School of Electrical and Computer Engineering and RMIT University.

Also, I would like to thank to my parents for all their support and encouragement during my research time.

Table of Contents

Table of Figures

Table of Tables

List of Abbreviations

AIS	artificial immune system
ANN	artificial neural networks
APCs	antigen presenting cells
B cell	bone marrow cell
BN	Bayesian network
CART	classification and regression trees
CE	classification Error
CNF	conjunction normal form
DBMS	database management system
DoS	denial of service
EA	evolutionary algorithms
EAI	Efficient AIS Based IDS
GA	genetic algorithm
GP	genetic programming
IDS	intrusion detection system
ICARIS	international conference on artificial immune systems
LAN	local area network
LGP	linear genetic programming
MARS	multivariate adaptive regression splines
MCE	mean classification error
R2L	remote to user
SIP	session initiation protocol
SVDF	support vector decision function ranking
SVM	support vector machines
T cell	thymus cell
U2R	user to root

1 Introduction

Driven by the rapid growth of the computer network technologies, the security of the computer and network information is becoming increasingly important. The appearances of the new access technologies and the advanced devices have increased the possibilities of malicious attacks or service abuses by various hackers. Also, with the appearances of multimedia services (video, audio, image, text, etc.), a faster, short-delay anti-virus system is required. However, the traditional passive defence mechanisms like encryptions and firewalls cannot fully meet current security requirements. Therefore, a special attack and misuse detection system is needed. The intrusion detection system (IDS) is such a system, which is composed by a series of devices and software applications to monitor network activities in order to protect the system from malicious activities.

The IDS can detect unauthorized users or processes by comparing the user behaviour with a user profile. Two approaches, misuse detection and anomaly detection, are usually used in the intrusion detection process. The misuse detection is used to detect the intrusion when the behavior of the system matches with any of the intrusion signatures in the user profile. And the anomaly detection, which is also called as outlier detection [1], is used to detect the intrusion when the given data set does not match with the established normal behavior.

Various techniques have been used for building IDS, like Support Vector Machines (SVM) [2], Multivariate Adaptive Regression Splines (MARS) [3], and Linear Genetic Programming (LGP) [4], etc. In recent years, the bio-inspired algorithms, such as Genetic Algorithm (GA) and Artificial Neural Networks (ANN), have been widely studied and applied in intrusion detection to reduce the labor costs and increase the system efficiency. In this thesis, the study is around Bio-inspired IDS technology. The focus is on applying Artificial Immune System (AIS) on IDS to improve the detection performance. AIS was first proposed in mid 1980s. Farmer, Packard and Perelson [5], Bersini and Varela's [6] work have started the area. AIS became a subject of its own in mid 90s. It has been defined by de Castro and Timmis [7] as: "Adaptive systems inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving." The early work of applying AIS to IDS can be found in [8]. A multilayer AIS based IDS was

proposed by Dasgupta [9] in order to provide systematic defence. These AIS based algorithms have achieved good detection results. But their computing complexity is quite high. In IDS, responding time is an important issue. The more complex the system, the more computing time and the longer responding time will be. Large parameter set in IDS can increase the detection accuracy. However, the more parameters used, the more complex the system. The trade-off between the complexity and the accuracy is a challenge.

An improved feature selection based IDS is proposed in this study. The anomaly detection of the IDS is set up based on AIS negative selection algorithm. And the feature selection algorithm is used to reduce the complexity of the system. A weight based rough set and LGP feature selection algorithm is proposed in our scheme. The main contribution of the rough set algorithm is its ability of reduction and the main advantage of the LGP algorithm is that it can be fast enough to detect real-time intrusions. It is assumed that different features have different contributions to the IDS performance, so that the coefficient weighting is introduced to improve the detection accuracy of the IDS. The higher contribution the feature to the system performance, the larger weight coefficient the feature will be given. A systematic testing has been done in our study to find out the best combination of the different features and their weight coefficients. The improved feature selection based IDS shows high abnormal detection rate and relatively low false alarm rate comparing with the other algorithms.

1.1 Scope

As it was observed, the traditional IDS has the following drawbacks: 1) high complexity due to the large normal behaviour profile; 2) relatively low detection accuracy; 3) low detection rate for the new patterns. In order to improve the IDS's complexity and detection accuracy, our research has focused on improving the following issues:

- What kind of algorithms should be used to reduce the system's complexity without sacrificing too much detection accuracy?
- How can we increase the IDS's detection accuracy?
- What algorithms should be used to detect the new malicious attack?

The research outcome is to design an improved bio-inspired based intrusion detection system to overcome the drawbacks mentioned above. The scope of this research includes:

- Review of current research on the AIS.
- An investigation of IDS concepts and technology used.
- Proposal of new AIS based IDS.
- Investigation of feature selection algorithms.
- Simulation development using C++.
    - Developing AIS based IDS.
    - Testing different feature selection environment on the IDS.
    - Choosing statistics and collecting results
- An in-depth performance analysis based on the comparison of the simulation results

1.2 Thesis Outline

The rest of the thesis was organized as follows: Chapter 2 describes the current network security issues and the background of the intrusion detection system. The drawbacks of the current network security system are discussed and the importance of the intrusion detection system is introduced. The artificial immune system is introduced in Chapter 3. The background and the current applications of artificial immune system are discussed. In Chapter 4, the data set used in this experiment is introduced. The data pre-processing and feature selection algorithm are also discussed in Chapter 4. The improved feature selection based IDS is proposed in Chapter 5. Different weight coefficient combinations are tested and discussed in this chapter. Chapter 6 gives the summary of this research. A possible future research works are discussed.

2 Network Security and Intrusion Detection Systems

2.1 Network Security

With the increasingly vital roles the network-based computer systems played in the modern society, the security of the systems becomes more important than ever before. The security of the system will be compromised while the intrusion happens. The intrusion is defined by Heady et al. [29] as "any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource". Several intrusion prevention techniques have already been used to protect computer systems as the first defence line like user authentication, data encryption and firewalls.

2.1.1 User Authentication

User authentication is used to verify the identity of users when they try to log onto a network in order to protect the computer security. Several techniques have been used to prove the identity of the user to the network like passwords, digital certificates, smart cards and biometrics. Authentication mechanisms differ in the number of verifiers they provide, e.g. single verifier supporting or multiple verifiers supporting. Also, mechanisms differ in the assurances they apply.

Sometimes, the traditional one factor password authentication is not enough due to the complexity of the current network environment. So, two-factor authentication is needed. The two-factor authentication requires two independent methods to establish the identity and the privileges, and it is also called strong authentication. In the two-factor authentication, the password is served as a secret word or code that is used to be a security measure against the unauthorized access and it is always managed by the operating system or DBMS.

2.1.2 Data Encryption

Data encryption has changed dramatically over the years, from the military solution only to widespread public use. The data encryption method is a fast, easy to use and secure way for network security. Two types of cryptography mechanisms can make the data invisible: secret-key and public-key.

For the secret-key cryptography system, if the sender wants to send a message to someone, it will first encrypt the message by using a secret-key, and the receiver will decrypt the encrypted message by using the same secret-key. This method is also known as symmetric cryptography. The main problem for the secret-key cryptography is that the sender and the receiver have to use a secure channel to exchange the secret-key and the secure channel is hard to find. For this reason, another cryptography system was invented called public-key cryptography system.

For the public-key cryptography system, each part will get two keys, public-key and private-key. The public-key is published to all the nodes in the system, and whoever wants to send a message to the receiver can use the receiver's public key. Then the receiver can decrypt the message by using its private-key. The RSA public-key cryptosystem is the most popular form of public-key cryptography.

The RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who proposed this algorithm first in 1978. It is an asymmetric encryption algorithm. A user of RSA needs to create the product of two large prime numbers, along with an auxiliary value served as the public key. After the public key is generated, the prime factors can be ruined. Anyone who wants to transfer information to the RSA user can use the public key to encrypt the message. The difficulty of factoring large integers determines the reliability of the RSA algorithm. There is not any reliable way to attack the RSA algorithm.

2.1.3 Firewalls

Firewall system is an important component of the network security to protect the network from the outside world. The firewalls provide strict accessing control between enterprise networks and Internet. All traffics pass through the firewalls will be monitored and controlled.

The common firewall architecture contains at least four components: the exterior router, an interior router, an exposed network and a secure server, each of which provides part of the complete security scheme. The architecture is shown in Figure 2-1.

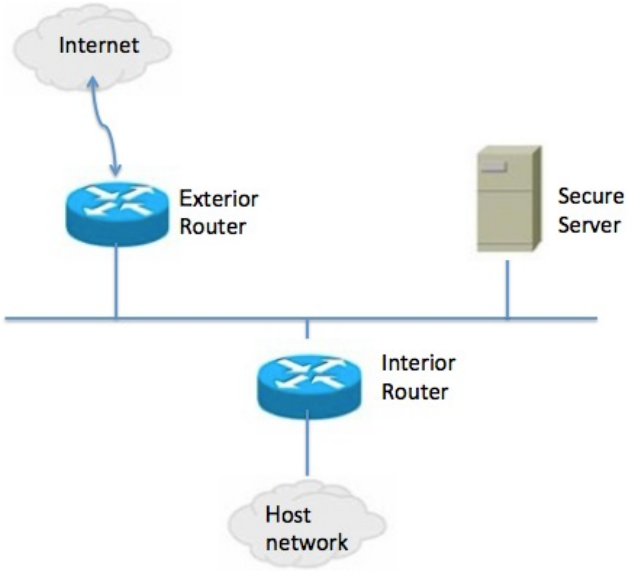

Figure 2-1 Basic firewall architecture

Several distinct advantages can be achieved by implementing the firewall into the network. Firstly, it is more secure to add a firewall than an average host. A second advantage comes from that the firewall machines have professional administration than the average system administrator. Also, the firewall, the gateway machine, need not be trusted by any other machines, which means even if the firewall has been compromised, no others will fall automatically.

Although firewall is a powerful tool for network security, it still has its limitations. One of these limitations is that a firewall has a very strong defence against attacks from lower level, while, for the problems from the higher level, it almost provide no protection. Also, the firewall can defence against the known flaws well, but for the new ones, it might be useless. So, it needs to be upgraded regularly.

2.2 Intrusion Detection Systems

For modern computer networks, the traditional intrusion prevention techniques are struggling as the system is becoming more and more complex. New efficient intrusion detection systems are needed as second layer defence. The main advantages of using intrusion detection systems include:

- Real-time reporting of break-ins in order to have timely countermeasures to protect important information.

- Identifying the trace of the intrusions so that the system administrators can identify the intruders and eliminate the security holes.

The most popular techniques to detect intrusions are done by using audit data that is generated by operating systems or by networks. For example, Lunt et al.[30] proposed the IDES which requires to use audit trails generate by a C21 or higher rated computer. There are also other techniques to detect intrusions like monitoring the network connections or information flows, like the Bro system proposed by Paxson [31].

The intrusion detection system can also help to analyse the audit data even after the attack happened in order to determine the extent of the damages occurred. The analysis is important because it will help for future prevention of such kind of attacks. This makes the intrusion detection systems not only a real-time detection mechanism but also a system-analysing tool.

The intrusion detection approach is classified into two types: misuse detection and anomaly detection.

2.2.1 Misuse Detection

The misuse detection is used to detect the intrusion when the behavior of the system matches with any of the intrusion signatures. The misuse detection system consists of the specific patterns of the known system vulnerabilities, then monitoring the current activities of such patterns. If the current activities match with any pattern in the alarm pattern database, an alarm will be raised. Most current misuse detection utilizes some form of the rule-based analysis. The NIDES system proposed by Lunt [32] can be used an example. It uses a rule-based algorithm for misuse detection. First, the system encodes the existing known vulnerabilities and attack types into rules. Then, the audit data will be compared with the rule conditions to determine whether an intrusion occurred or not. For the rule-based approach, a comprehensive rule set is critical in the application of expert systems for intrusion detection.

The key advantage of the misuse detection is that it can detect the known attacks accurately and rapidly. Unfortunately, by the nature of misuse detection, it is not very effective in detecting innovative attacks. So, the misuse detection system requires updating frequently. The lack of update will degrade the security level of the entire

system. The situation of current networks is that new attack techniques are appearing more and more frequently, so the misuse intrusion systems may need to be updated across many platforms more often than ever before. This will cause very labor intensive for constructing and maintaining a misuse detection system.

2.2.2 Anomaly Detection

The anomaly detection is used to detect the intrusion when the given data set does not match with the established normal behavior. The anomaly detection consists of a normal behavior profile that includes the specific normal patterns of the network, then the monitored patterns will be compared with the normal behavior profile. If the monitored patterns do not conform to the normal behavior profile, these non-conforming patterns will be referred to as anomalies. Although the approach of anomaly detection looks straightforward, there are still several factors make it challenge:

- It is very difficult to build a normal behavior profile that contains every possible normal behavior.
- The boundary between normal and abnormal behavior is not precise in some time.
- The malicious adversaries will often adapt themselves to appear like normal, so that it makes it more difficult to distinguish.
- The normal behavior might keep evolving, so the current knowledge of the normal behavior might not be sufficiently enough to represent all the normal behaviors in the future.

Due to the challenges mentioned above, the anomaly detection system is not easy to build. Therefore, the current existing anomaly detection techniques will always focusing on a specific area like anomaly detection for IP networks, or anomaly detection for SIP networks.

However, the anomaly detection systems do have their own advantages. They can detect unknown intrusions because a priori knowledge about specific intrusions is not required. Also, by adding the statistical-based algorithm, the anomaly detection system will be adaptive to the changing circumstance because it is relatively easier to update the statistical measures.

Numeric researches have been focused on the anomaly detection area due to its ability to detect unknown intrusions. Patcha and Park [33] present a survey of current anomaly detection techniques focusing on the computer network intrusion detections. NIDES proposed by Lunt [32] is an anomaly detection system that uses a statistically measured user normal profile. Zainal et al. [34] introduced the feature selection algorithm into the anomaly detection system.

2.2.3 Types of Network Intrusion

In practice, the network intrusion includes six types.

1. Misuse/abuse: misuse or abuse happens when unauthorized activities done by authorized users. For example, the theft of information.
2. Reconnaissance: reconnaissance occurs when an intruder try to determine whether the system or services can be exploitable.
3. Penetration attempt: penetration attempt occurs when the unauthorized activity tries to gain access to the computing resources.
4. Penetration: penetration occurs when unauthorized users successfully access to the computer resources.
5. Trojanization: trojanization happens when unauthorized processes can present and active.
6. Denial of service (DoS): denial of service happens when attacks influence the legitimate users to access the computing resources.

2.2.4 Terminology of the IDS

In order to describe the performance of the intrusion detection system, several terminologies are introduced:

- Alarm: A signal suggesting that a system has been or is being attacked.
- True Positive: A legitimate attack that triggers an IDS to produce an alarm.
- False Positive: An event signaling an IDS to produce an alarm when no attack has taken place.
- False Negative: A failure of an IDS to detect an actual attack.
- True Negative: When no attack has taken place and no alarm is raised.
- Noise: Data or interference that can trigger a false positive.

- Site policy: Guidelines within an organization that control the rules and configurations of an IDS.
- Site policy awareness: An IDS's ability to dynamically change its rules and configurations in response to changing environmental activity.
- Confidence value: A value an organization places on an IDS based on past performance and analysis to help determine its ability to effectively identify an attack.
- Alarm filtering: The process of categorizing attack alerts produced from an IDS in order to distinguish false positives from actual attacks. [35]

2.2.5 Current Intrusion Detection Systems Issues

The quality of an IDS can be measured by its effectiveness, adaptability and extensibility. For the effectiveness issue, the IDS should have high true positive rate and low positive rate (false alarm rate). For the adaptive issue, the IDS should have the ability to detect both known intrusions and can quickly adapt to innovation intrusions.

Because of the increasing complexity of the modern networks, the current expert knowledge based IDS are usually incomplete and not precise enough. The trend of focusing on solving the “current” vulnerabilities makes the current IDS unable to detect “future” attacks. In order to overcome these drawbacks, an AIS based IDS is proposed. By introducing the artificial immune systems into the IDS, the adaptive issue can be solved by using the negative selection algorithm. The immunology algorithms help to increase the systems’ effectiveness.

3 Artificial Immune System

This chapter provides a brief introduction of the artificial immune system. The artificial immune system is a branch of bio-inspired computational intelligence, and it has attracted increasingly interest from the researchers after it was first proposed. A number of researches have been focused on the AIS area, ranging from modeling natural immune systems, solving artificial related problems, to anomaly detection, and controlling.

3.1 A Brief History of AIS

Artificial Immune System (AIS) is a various area of researches that attempt to build a bridge between immunology and engineering by using the techniques like mathematical and computational modeling of immunology.

The origin of AIS is rooted in the early theoretical work of J.D. Farmer, N.H. Packard, A.S. Perelson [10, 11] and F. Varela, A. Coutinho, B. Dupire, N. Vaz [12]. It was first proposed in mid 1980s and became a subject of its own in mid 90s. Originally, AIS was aimed to find efficient abstractions of processes in the immune system,e.g [13] whilst by carefully reviewing the successful of this efficient natural mechanism, an increasing number of computer scientists proposed artificial immune based computer models to solve various problems ranging from virus detection, fault analyzing to clustering. Two researchers played an important role in crossing the divide between computing and immunology who are Hugues Bersini and Stephanie Forrest. Bersini and Forrest did a lot of basic works rooted from immunology and their works formed a solid foundation of the area of AIS. With regards to Bersini, he was focusing on the basic theory of immune network and examining how the immune system maintained its memory and how to build a model to mimic that progress. And for Forrest, she was focusing on the application area of the AIS. She proposed the idea of introducing the immune system into the computer security area by using its ability to distinguish between self and non-self. AIS has been defined by Castro and Timmis [14] as: “Adaptive systems inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving.”

The International Conference on Artificial Immune Systems (ICARIS) was started in 2002 due to the growing amount of work conducted on the AIS area, and the conference series continues to operate to now. The ICARIS aims to provide a forum for AIS researchers to present and discuss their latest advances.

3.2 Biological underpinnings of AIS

Three main immunological theories have been focused by AIS, which are negative selection, clonal selection and immune networks. AIS researchers focus on clonal selection and immune networks because of their inherent learning and memory mechanisms. And for negative selection, researchers are interested in the generation of detectors that are capable to classify changes in self. The negative selection, clonal selection and immune networks will be detail discussed in the following section.

3.2.1 Negative Selection Algorithm

The important character of the human immune system is that it can maintain its diversity and generality, and it can detect a large number of antigens by using a small number of antibodies. In order to make it possible, several functions will be processed [15]. One of those functions is to develop the antibodies through the gene library. The gene library will be used in creating thymus cell (T cell) and bone marrow cell (B cell). While creating a new antibody, the gene segments in the gene library will be randomly selected and assembled. As shown in Figure 3-1, large number of antibodies can be generated from combining different gene segments in the gene library.

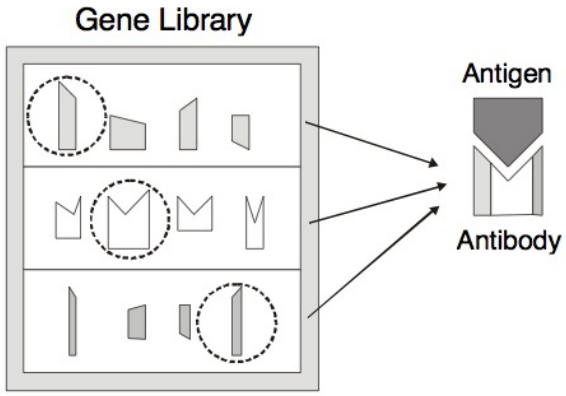

Figure 3-1 Gene Expression Process [16]

However, one problem will appear due to the full immune response above. As well as responding to harmful antigens, those newly generated antibody may also react to self-cells coming from the host. In order to protect the body from self-reactive, the human immune system produces the negative selection. The negative selection of T cells occurs within the thymus. After the immature T cell is generated, it will be compared with the self-peptide presented by antigen presenting cells (APCs). Those that bind to self-peptide (have high affinity) will be eliminated through a controlled death. Only those do not have self-reactive and can recognize antigens will be distributed throughout the human body. Therefore, this process forms the foundation of the AIS work.

By observing the similarities between the elimination harmful antibodies in human body and the anomaly detection, Forrest, Hofmeyr, and Somayaji [17] proposed the negative selection algorithm for anomaly detection. The artificial negative selection is focusing on the generation of the detectors and the detectors are intended to use to detect anomaly patterns. The algorithm of artificial negative selection is as follows,

1. Create a set of self strings (S).
2. Randomly generated a set of strings (R).
3. For those $r_0 \in R$, if under certain matching algorithms, r_0 does not match with any $s \in S$, then, r_0 will be saved in the detector set (D). Otherwise, r_0 will be rejected.

The steps above can be described in Figure 3-2.

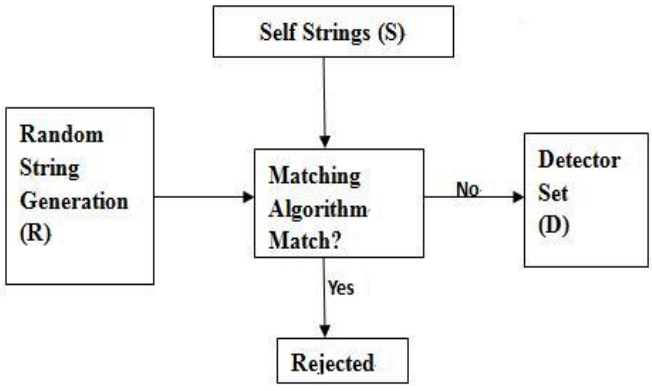

Figure 3-2 The principle of negative selection algorithm

As shown in Figure 3-2, the matching algorithm is the core of the negative selection. The affinity between the random string R and self-string S is decided by using a matching algorithm. Several algorithms have been proposed in this area to determine the affinity. Among the popular ones are Eucliden algorithm, hamming distance algorithm and r-contiguous bit rule algorithm. The Affinities of the R and S are related to the distance between them. The definition of the distance can be shown as follows,

Let R=<r_0, $r_1 \ldots r_m$>, S=<s_0, $s_1 \ldots s_m$>,

- Eucliden $D = \sqrt{\sum_{i=0}^{m} (r_0 - s_0)^2}$ (3-1)

- Manhattan $D = \sum_{i=0}^{m} |r_0 - s_0|$ (3-2)

- Hamming $D = \sum_{i=0}^{m} \delta$, where $\delta = \begin{matrix} 1 & \text{if} \quad r_0 \neq s_0 \\ 0 & otherwise \end{matrix}$ (3-3)

For the negative selection, if the sum of distance between R and the S is equal or lager then a threshold, the R will be eliminated. Otherwise, the R will be kept as a detector.

3.2.2 Clonal Selection Algorithm

The clonal selection theory is used to explain the response to an antigenic stimulus in the adaptive immune system. It performs well in the computational optimization and pattern recognition areas (e.g. B Cell Algorithm [19]).

The idea is that only those cells capable of recognizing an antigen will proliferate, while others that do not recognize an antigen are selected against [18]. When a human body is exposed to an antigen, the bone marrow derived cells (B cell) will respond by producing antibodies (Ab). By binding with these antibodies, the antigen will stimulate the B cell to proliferate. In the B cell cloning process, they will undergo somatic hypermutation to introduce diversity into the B cell population. From the computational point of view, two important features in the B cell's affinity maturation can be exploited. The first one is that the B cell's proliferation is proportional to the affinity of the antigen that binds it, which means the higher the affinity is, the more clones will be produced. Secondly, the mutations suffered by the antibody of a B-cell are inversely proportional to

the affinity of the antigen it binds. By applying these two features, de Castro and Von Zuben [20] proposed the clonal selection based AIS called CLONALG. de Castro and Timmis use the CLONALG to perform pattern matching and multi-modal function optimization in [21].

3.2.3 Immune Networks Algorithm

In 1974, Niels Kaj Jerne proposed an immune network theory in the landmark paper [22]. The immune network algorithm helps to explain some of the observed emergent properties of the immune system like learning and memory. The algorithm is focusing on the network graph structures involved where antibodies represent the nodes, and the training algorithm involves growing or pruning edges between the nodes based on affinity. The immune networks algorithm performs well in the area of clustering, data visualization, controlling, and optimization domains.

The algorithm aiNet, inspired by the immune network theory, is one of the most popular clustering algorithms. The aiNet was developed by de Castro and von Zuben [23] in order to find a reduced set of points that closely represents the input set of points. In aiNet, immune learning and memory are consequences of network interactions and antigenic stimulation, also bridging a gap between the originally conflicting theories of clonal selection and idiotypic networks.

3.3 AIS Applications

As a new paradigm, there have been a lot of successful applications in AIS. By comparing with other existing bio-inspired paradigms like Evolutionary Algorithms (EA), Neural Networks (NN), and other traditional classification or clustering algorithms, AIS performs well in several fields (Table 3-1). The categories shown in Table 3-1 can be briefly summarized in three big domains: learning, anomaly detection and optimization. Those three domains can map to the three main algorithms in Section 3.2. In the Learning domain, it includes clustering/classification, robotics, controlling and pattern recognition applications. Anomaly detection domain includes virus detection and computer and network security applications. The optimization domain contains numeric function optimization and real world problems especially on the combinatoric area.

Table 3-1 Application areas of AIS

Major	Minor
Clustering/ Classification	Bio-informatics
Anomaly Detection	Image Processing
Computer Security	Robotics
Numeric Function Optimisation	Controlling
Combinatoric Optimisation	Virus Detection
Learning	Web Mining

In this project, the focus is on the anomaly detection area using AIS, which is a hot research area. For anomaly detection, it requires to decide whether an unknown pattern reflects a normal behavior or an intrusion. The AIS based anomaly detection relies on a detector set which is created using a known normal pattern set (or class) beforehand. The live IDS is expected to detect any intrusion pattern when it occurs.

The pioneering works done by Forrest, Perelson and Allen [24] led to a great deal of research and proposal of immune inspired anomaly detection systems [25]. These works hint the possibility that the immune based approach might be useful to solve the intrusion detection problem to some degree. More work has been done by Kim and Bentley [26], which use the clonal selection algorithm and negative selection algorithms together to reduce the false positive rate. Also, several researchers have focusing on improving the matching algorithms in order to improve the performance of the system. Balthrop et al. [27] proposed the r-chunk matching rule to replace the r-contiguous bits matching rule in order to reduce the computational complexity. D'haeseleer, Forrest and Helman [28] showed several advantages of AIS based anomaly detection. One of the most important one is that the AIS based anomaly detection does not define specific anomalies to be detected, which means it does not require the pre-knowledge of the anomalies. This feature allows the AIS based anomaly detection system to be able to detect the previously unseen anomalies.

Still, the AIS based anomaly detection has some drawbacks like scaling issues, high false positive rates and also complexity issues. Stibor, Timmis, and Eckert [36] proposed that on testing high-dimensional data set (such as KDD Cup 99), the negative selection has very poor performance comparing with other techniques. But when the problem size scales down, the performance impact factor will decreases to almost zero. Some researchers have proposed the danger theory approach to overcome some of these drawbacks [37][38]. They assume that the traditional methods in determining what is 'normal' for a system should be moved away, and a dynamically identifying 'normal' should be used through the adoption of danger signals.

4 Efficient AIS Based IDS (EAI)

As discussed in Chapter 3, the negative selection used in AIS has its intrinsic advantages in dealing with network intrusion detection problems. In this project, the exploration on AIS based intrusion detection system is to further improve the IDS's performance and system complexity.

AIS based intrusion detection system can be split into the following steps:

1. Normal Data Set Gathering: In this step, the data set of parameters on normal behaviour of the system will be collected. These data will be used in the negative selection process.
2. Detector Generation: In this step, the detectors of the IDS will be generated based on the negative selection algorithm and stored in the system memory.
3. Live Detection: The detectors of the IDS will be used in monitoring the networks. The network live data will be compared with the detectors in order to detect abnormality.

Before establishing the intrusion detection systems, some fundamental issues should be considered, e.g. what kind of network environment our IDS will work in, or how we test our IDS, or what kind of data do we want to use. All of these issues are related to the chosen of experiment data.

4.1 Experiment Data

Generally speaking, two approaches can be used to gather the experiment data, each of which has its advantages and disadvantages.

The first method is to gather the experiment data from an experimental environment where a virtual network is created and network packet capturing tools can be used to obtain the monitoring parameters. Several researchers have used this method. Kim and Bentley [16] used *tcpdump* as the data packet-capturing tool in their experiment. The TCP packet passed between the intra-LAN and external networks are all collected in their scheme. All the data collected by the *tcpdump* will have the *tcpdump* format such as time stamp, source and destination IP address, source and destination port number etc. Martin Roesch developed the snort, an open-source IDS, based on the *tcpdump.*

The advantage of this method is that the researchers can collect as many data as they want for the training or testing purposes. But the disadvantages are also obvious. The laboratory virtual networks are not real-world networks so that the IDS might perform well in the laboratory environment but it might not fit for the real networks. Also, it is hard for the researches to compare their IDS with other IDS because the different testing environment they use.

The second method is to use the existing data set provided by some laboratory specific for network intrusion detection issue. The most widely used data set for intrusion detection is the Knowledge Discovery and Data mining 1999 data set (KDD Cup 99). The KDD Cup 99 data set contains a rich set of different attack data types and normal data, which can fulfill the training and testing purpose for researches. Betanzos et.al.[40], Shyu et.al.[41], and Saravanan [42] are all use KDD Cup 99 data set as their training and testing data.

The advantage of using the existing well-accepted data set is that the researchers can easily compare their IDS performance with others so that the result of the experiment will be more convincing. Also, it will save a lot of time in gathering data. Unfortunately, by using the existing data set, it is hard for the researchers to add new data, and they have to wait the upgrade package from the laboratory.

By comparing the two methods, the existing well-accepted data set has more advantages as the performance of the IDS can be easily compared with other schemes. Therefore, the KDD Cup 99 data set is selected in our experiment.

4.1.1 The KDD Cup 99 Data Set

The KDD Cup 99 data set is the most widely used data set for network-based intrusion detection. This data set is built based on the data captured in DARPA'98 IDS evaluation program [43]. The DARPA'98 Intrusion Detection Evaluation Program was managed by MIT Lincoln Labs. It aims to survey and evaluate the researches in the intrusion detection area. Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. They operated the LAN as if it were a true Air Force environment, but peppered it with multiple attacks [44]. The data set contains 24 training attack types and 14 additional attack types in the test data only. These attacks fall into four main categories:

1. **Denial of service (DOS)**: In this type of attack, an attacker makes some computing or memory resources too busy or too full to handle legitimate requests, or denies legitimate users access to a machine. Examples are Apache2, Back, Land, Mailbomb, SYN Flood, Ping of death, Process table and Smurf.
2. **Remote to user (R2L)**: In this type of attack, an attacker who does not have an account on a remote machine sends packets to that machine over a network and exploits some vulnerability to gain local access as a user of that machine. Examples are Dictionary, Ftp_write, Guest, Imap, Named, Phf, Sendmail and Xlock.
3. **User to root (U2R)**: In this type of attacks, an attacker starts out with access to a normal user account on the system and is able to exploit system vulnerabilities to gain root access to the system. Examples are Eject, Loadmodule, Ps, Xterm, Perl and Fdformat.
4. **Probing**: In this type of attacks, an attacker scans a network of computers to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use this information to look for exploits. Examples are Ipsweep, Mscan, Saint, Satan, Imap.

For each record in the KDD Cup 99 data set, it contains 41 parameters and a data type at the end of each record, shown as follows:

- 0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00,0.11,0.00,0.00,0.00,0.00,0.00,normal.
- 0,icmp,ecr_i,SF,1032,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,511,511,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,255,1.00,0.00,1.00,0.00,0.00,0.00,0.00,0.00,smurf.
- 0,tcp,http,SF,239,486,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,19,19,1.00,0.00,0.05,0.00,0.00,0.00,0.00,0.00,normal.

Each parameter, separated by the comma in the data string, has its own meaning and it is shown in Table 4-1.

Table 4-1 The KDD Cup 99 parameters

Parameter	Description	Parameter	Description
1.Duration	Length (number of seconds) of the connection.	2.Protocol_type	tcp,udp,icmp

3.Service	Network service on the destination.	4. Flag	Normal or error status of the connection.
5.Src_bytes	Number of data bytes from source to destination.	6. dst_bytes	Number of data bytes from destination to source.
7. land	1 represents that the connection is from/to the same host/port. 0 otherwise.	8.wrong_fragment	Number of wrong fragments.
9. urgent	Number of urgent packets.	10. hot	Number of "hot indicators".
11.num_failed_login	Number of failed login attempts.	12.logged_in	1 represents successfully logged in, 0, otherwise.
13.num_compromised	Number of "compromised" conditions.	14. root_shell	1 represents root shell is obtained. 0, otherwise.
15.su_attempted	1 if "su root" command attempted; 0 otherwise.	16. num_root	Number of "root" accesses.
17.num_file_creations	Number of file creation operations.	18. num_shells	Number of shell prompts.
19. num_access_files	Number of operations on access control files.	20.num_outbound_cmds	Number of outbound commands in an ftp session.
21.is_hot_login	1 if the login belongs to the "hot" list; 0 otherwise.	22.is_guest_login	1 if the login is a "guest" login; 0 otherwise
23.count	Number of connections to the same host as the current connection in the past 2 seconds.	24.srv_count	Number of connections to the same services as the current connection in the past 2 seconds.
25.serror_rate	% of connections that have "SYN" errors.	26.srv_serror_rate	% of connections that have "SYN" errors.
27.rerror_rate numeric	% of connections that have "REJ" errors.	28.srv_rerror_rate	% of connections that have "REJ" errors.
29.same_srv_rate	% of connections to the same service.	30.diff_srv_rate	% of connections to different services.
31.srv_diff_host_rate:	% of connections to different hosts.	32.dst_host_count	number of connections from the same host to destination during a specified time window.
33.dst_host_srv_count	Count of connections having the same destination host and using the same service	34.dst_host_same_srv_rate	% of connections having the same destination host and using the same service
35.dst_host_diff_srv_rate	% of different services on the current host	36.dst_host_same_src_port_rate	% of connections to the current host having the same src port
37.dst_host_srv_diff_host_rate	% of connections to the same service coming from different hosts	38.dst_host_serror_rate	% of connections to the current host that have an S0 error
39.dst_host_srv_serror_rate	% of connections to the current host and specified service that have an S0 error	40.dst_host_rerror_rate	% of connections to the current host that have an RST error
41.dst_host_srv	% of connections from the same host		

_rerror_rate	with same service &REJ errors to the destination host during a specified time window.		

4.2 Data Preprocessing

Although the KDD Cup 99 data set has already undergone an initial data pre-processing comparing with the raw *tcpdump* format provided by DARPA data, some data pre-processing is still needed in this experiment. The KDD Cup 99 data set contains 34 numerical features and 7 symbolic features. For the numerical features, they can be used straightway. For the symbolic parameters, they need to be transformed into numerical ones. In those seven symbolic features, four of them (land, logged_in, is_hot_login, is_guest_login in Table 4-1) have only two categories, so a binary conversion is enough. The other three symbolic features (protocol, service, flag in Table 4-1) have more than two categories: 3 categories for protocol, 64 categories for service, and 11 categories for flag. In our experiment, they are mapped to numerical values ranging from 0 to N-1 where N is the number of symbols. Taking the "flag" parameter for example, the C++ code for mapping the 11 categories of flag parameter is shown as follows:

```
        s_mapStringValues["S0"]=connection_S0_Value0;
s_mapStringValues["S1"]=connection_S1_Value1;
s_mapStringValues["SF"]=connection_SF_Value2;
s_mapStringValues["OTH"]=connection_OTH_Value3;
s_mapStringValues["S2"]=connection_S2_Value4;
s_mapStringValues["RSTO"]=connection_RSTO_Value5;
s_mapStringValues["S3"]=connection_S3_Value6;
s_mapStringValues["RSTR"]=connection_RSTR_Value7;
s_mapStringValues["RSTOS0"]=connection_RSTOS0_Value8;
s_mapStringValues["SH"]=connection_SH_Value9;
s_mapStringValues["REJ"]=connection_RSTRH_Value10;
```

For our experiment, the only parameter quantization is not sufficient enough. As discussed in Section 3.3, Stibor, Timmis, and Eckert [36] showed that the negative selection would have very poor performance on high-dimensional data set. While for KDD Cup 99 data set, they have 42 parameters, which mean 42 dimensions. So, the feature selection mechanism is needed to reduce the dimension of the experiment data set. Also, by using the feature selection mechanism, the system complexity will be reduced in the training progress.

4.2.1 Feature Selection

In modern networks, the number of audit data that an intrusion detection systems need to monitor is increasing even in small networks. The analysis is increasingly difficult even with high-speed computers. Therefore, the IDS needs to reduce the amount of data to be monitored, which is extremely necessary in real-time detection. The raw data reduction can be achieved in several steps. At first, the data that are not considered as useful can be filtered. This step is called data filtering. While in this experiment, this step is not being considered because the KDD Cup 99 data set is already filtered by the Lincoln Lab. The feature selection is to further eliminate some parameters from the data set. A feature set will be formed after the feature selection process. The aim is to reduce the system complexity while maintaining a good IDS performance.

In high-dimensional feature domains, some features may have negative correlations and that will hinder the process of AIS-based intrusion detection. Also, some information is reflected in several features and these features have redundant information. The more features input into the IDS, the more computation complex the system will be. Features selection in IDS is to select a subset of the features. For distributed IDS, the feature selection should analyze the network related information such as source and destination IP addresses, protocol type, duration of the connection, number of data bytes from source to destination etc. All these features can be found in the KDD Cup 99 data set (see Table 4-1). The difficulty of applying feature selection in IDS is to identify which of these features are irrelevant or act as a redundancy for IDS, and which of these features have contribution or essential for IDS. Even in the case that there are no useless features in the raw data, by selection the most important ones might well improve the responding time of IDS and maintain the detection accuracy.

The feature selection problem for IDS that use the KDD Cup 99 data set can be characterized in the following:

- The large number of input data $x = (x_1, x_2, x_3, \ldots, x_{41})$ have varying degrees of impact on the output of the intrusion detection system y.
- It is assumed that there should be a mathematical formula that can describe the relationships between input data x and output y, $y = F(x)$.

- By using the large number of data that provided by the KDD Cup 99 data set, several important features for the intrusion detection system will be exploited.

Till now, there is no model or function that can precisely describe the relationships between different attacks and features. Many researchers have focused on feature selections on IDS. They use the data mining techniques for feature selection. Sung and Mukkamala[45] introduced three techniques in the IDS feature selection that are Support Vector Decision Function Ranking (SVDF), Linear Genetic Programming (LGP) and Multivariate Regression Splines (MARS). Zainal et al. [34] and Zhang et al. [46] proposed the feature selection using rough set in intrusion detection. Chebrolu et al. [47] used the Bayesian Network (BN) and Classification and Regression Trees (CART) as the feature selection tool to solve the real-time IDS issues.

The work done by the previous researchers show that there are features that have significant influence on the IDS performance. And by using different types of feature selection algorithms, the selected features from the same data set might be different. There are still some problems in the works mentioned above. The most important one is that in all the reported works, they just picked up several most important features from the specific data set, and they assumed that all those important features would have equally contributions on the IDS performance. For example, Chebrolu et al. [47] choose four features and Sung and Mukkamala[45] choose six features in testing their IDS. According to the feature selection theory, for the reason that different features have different contributions on the system output, they should be given different weight. In this project, an improved feature selection algorithm using rough set and LGP separately is proposed to enhance the performance of an AIS based IDS.

4.2.1.1 Rough Set Algorithm (RSA)

Rough set algorithm was proposed by the Polish computer scientist Zdzislaw Pawlak in 1982 [48]. It is an extension of conventional set theory and a mathematical tool that support approximations in decision making. Also, it suits well for classification of objects. The rough set is an approximation of a vague set by a pair of precise concepts, which are called lower and upper approximations. The lower approximation is a description of the domain objects that are known with certainty to belong to the subset of interest, whereas the upper approximation is a description of the objects that possibly

belong to the subset [49]. The main contribution of the rough set theory is its ability of reduction. It can provide a minimal subset of attributes that contains the same capability of objects classification as the whole set of attributes. The following definitions can show how the reductions obtained [48].

Definition 1 An information system is defined as a four-tuple as follows, $S =< U, Q, V, f >$, where $U = \{x_1, x_2, \ldots, x_n\}$ is a finite set of objects (n is the number of objects); Q is a finite set of attributes, $Q = \{q_1, q_2, \ldots, q_n\}$; $V = \cup_{q \in Q} V_q$ and V_q is a domain of attribute q; $f: U \times Q \to V$ is a total function such that $f(x, q) \in V_q$ for each $q \in Q, x \in U$. If the attributes in S can be divided into condition attribute set C and decision attribute set D, i.e. $Q = C \cup D$ and $C \cap D = \Phi$, the information system S is called a decision system or decision table.

Definition 2 Let $IND(P)$, $IND(Q)$ be indiscernible relations determined by attribute sets P, Q, the P positive region of Q, denoted $POS_{IND(P)}(IND(Q))$ is defined as follows:

$$POS_{IND(P)}(IND(Q)) = \cup_{X \in U/IND(Q)} IND(P)\_(X) . \quad (4\text{-}1)$$

Definition 3 Let P, Q, R be an attribute set, we say R is a reduct of P relative to Q if and only if the following conditions are satisfied:

1) $\mathrm{POS}_{\mathrm{IND(R)}}(\mathrm{IND(Q)}) = \mathrm{POS}_{\mathrm{IND(P)}}(\mathrm{IND(Q)})$; (4-2)
2) $\forall r \in R$ follows that $POS_{IND(R-\{r\})}(IND(Q)) \neq POS_{IND(R)}(IND(Q))$ (4-3)

Definition 4 Let $L = (U, A \cup \{d\}, V, f)$ be a decision system, whose discernibility matrix $M(U) = [M_A^d(i,j)]_{n \times n}$ is defined as:

$$M_A^d(i,j) = \begin{cases} \{a_k | a_k \in A \wedge a_k(x_i) \neq a_k(x_j)\}, d(x_i) \neq d(x_j) \\ \emptyset \qquad\qquad\qquad\qquad\qquad d(x_i) = d(x_j) \end{cases} \quad (4\text{-}4)$$

where $a_k(x_j)$ is the value of objects x_j on attribute a_k, $d(x)$ is the value of object x on decision attribute d.

Write $M(U) = [M_A^d(i,j)]_{n \times n}$ as a list $\{p_1, \ldots, p_t\}$. Each p_i is called a discernibility entry, and is usually written as $p_i = a_{i1}, \ldots, a_{im}$, where each a_{ik} corresponds to a condition attribute of the information system, k=q,...,m; i=1,...,t.

Furthermore, the discernibility matrix can be represented by the discernibility function *f*, conjunction normal form (CNF), i.e., $f = p_1 \wedge \ldots \wedge p_t$, where each $p_i = a_{i1} \vee \ldots \vee a_{im}$ is called a clause, and each a_{ik} is called an atom. Note that the discernibility function contains only atoms, but not negations of atoms. Although the discernibility matrix and discernibility function have different styles of expression, they are actually the same in nature.

Definition 5 Let h denote any Boolean CNF function of m Boolean variables $\{a_1^*, \ldots, a_m^*\}$, composed of n Boolean sums $\{s_1, \ldots, s_n\}$. Furthermore, let $w_{ij}^* \in \{0,1\}$ denote an indicator variable that states whether a_i^* occurs in s_j. $s_j = \sum_{i=1}^{m} w_{ij}^* \times a_i^*$, $h = \prod_{j=1}^{n} s_j$. We can interpret h as a bag or multise $\boldsymbol{M}(h) = \{S_i | S_i = \{a_j \in A | a_j^* \; occurs \; in \; s_i\}\}$. Because the discernibility function f is also a CNF Boolean function, so it has a multiset. Let $\boldsymbol{M}(f)$denote the multiset of discernibility function f, $\boldsymbol{M}(f) = \{\{a_{11}, \ldots, a_{1m}\}, \ldots, \{a_{i1}, \ldots, a_{im}\}, \ldots, \{a_{t1}, \ldots, a_{tm}\}\}$.

Definition 6 A hitting set of a given bag or multiset M of elements from 2^A is a set $B \subseteq A$ such that the intersection between B and every set in M is nonempty. The set $B \in HS(S)$ is a minimal hitting set of M if B ceases to be a hitting set if any of its elements are removed. Let $HS(M)$ and $MHS(M)$ denote the sets of hitting sets and minimal hitting sets, respectively, $HS(M) = \{B \subseteq A | B \cap S_i \neq \Phi \; for \; all \; S_i \; in \; M\}$.

Definition 7 A approximate hitting set is a set that hits "enough" elements of the bag or multiset M. The approximate hitting set provides an approximate solution to the hitting set problem. The set of ε-approximate hitting sets of the multiset M is denoted $AHS(M, \varepsilon)$:

$$AHS(M, \varepsilon) = \left\{ B \subseteq \mathrm{A} \middle| \frac{S_i \; in \; M | S_i \cap B \neq \emptyset}{|M|} \geq \varepsilon \right\}, \tag{4-5}$$

where the parameter ε controls the degree of approximation. The set is a minimal ε-approximate hitting set if it ceases to be so if any of its elements are removed. The set of all minimal ε-approximate hitting set is denoted $MAHS(M, \varepsilon)$.

Definition 8 The significance of attribute is defined as: $SGF(a, R, D) = p(a)$, $p(a)$ is the number of appearing times of attribute a in the remain part of the discernibility matrix which removes all the elements that have nonempty intersection with R.

The work of Zainal et al. [34] has shown that the features set of the RSA in the KDD Cup 99 data set are service, flag, src_byte, srv_count, dst_host_count, and dst_host_srv_rerror_rate. And the performance of the intrusion detection system using rough set is shown in Table 4-2.

Table 4-2 IDS performance using RSA feature selection [34]

Type		Detection Accuracy
Normal		89.84%
Attack	DoS	99.34%
	Probe	99.63%
	U2R	100%
	R2L	100%
	Mean	99.743%

As shown in Table 4-2, the rough set based IDS performs well in detecting the attack on the system. But the false alarm rate is really high, which needs further investigation.

4.2.1.2 Linear Genetic Programming (LGP)

LGP is a variant of the Genetic Programming (GP) technique that acts on linear genomes [50]. The main characteristics of LGP in comparison to the tree-based GP lies in that the evolvable units are the expressions of an imperative language like C or C++, but not the functional programming language like LISP. The LGP selection procedure can put the lowest selection pressure on the individuals by allowing only two individuals to participate in a tournament [45]. The winner copy will replace the loser of each tournament, so that the crossover points will only occur between instructions. While inside the instructions, the instruction identifier will be randomly replaced by the mutation operation. The maximum size of the program in LGP is usually restricted in order to prevent programs without bounds. For the reason that the LGP can be implemented at machine code level, it can be fast enough to detect real-time intrusions.

Sung and Mukkamala [45] indicate that for the LGP feature selection algorithm, the main focus is in representation of the space of all possible subsets of the given input feature set. The feature in the candidate feature set is considered as a binary gene. Each individual consisting of fixed-length binary string represents some subset of the given feature set. For example, an individual with length d corresponds to a d-dimensional binary feature vector Y. Each bit of the individual represents the elimination or inclusion of the associated feature. $y_i = 0$ indicates elimination and $y_i = 1$ means inclusion of the i^{th}feature. Fitness F of an individual program p is calculated as the mean square error (*MSE*) between the predicted output (O_{ij}^{pred}) and the desired output (O_{ij}^{des}) for all n training samples and m outputs [51].

$$F(P) = \frac{1}{n*m}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(O_{ij}^{pred} - O_{ij}^{des}\right)^2 + \frac{w}{n}CE = MSE + w * MCE, \quad (4\text{-}6)$$

where Classification Error (*CE*) is computed as the number of misclassifications. Mean Classification Error (*MCE*) is added to the fitness function while its contribution is proscribed by an absolute value of weight (W).

The work of Sung and Mukkamala [45] indicated that the feature set of LGP in the KDD Cup 99 data set are service, src_byte, logged_in, rerror_rate, srv_diff_host_rate, dst_host_diff_srv_rate. The performance of the intrusion detection system using LGP is shown in Table 4-3.

Table 4-3 IDS performance using LGP

Type		Detection Accuracy
Normal		94.16%
Attack	DoS	99.8%
	Probe	100%
	U2R	60%
	R2L	100%

	Mean	89.950%

4.3 Efficient AIS Based IDS

4.3.1 Normal Data Gathering Step

As mentioned in Section 4.1, the data used in this project is the KDD Cup 99 data set. For each single record in the data set, it contains 41 parameters. In our experiment, the feature selection algorithm is used (Section 4.2.1) to simplify the raw data in order to reduce the system complexity. So, the useful parameters need to be extracted from the 41 parameters. Also, for the reason that the negative selection algorithm is used as the detector generation algorithm, the only data that need to collect are the normal behaviour data.

The raw data in the KDD Cup 99 data set is like follows:

- 0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00,0.11,0.00,0.00,0.00,0.00,0.00,normal.

The last pattern of each record in the data set represents its type (normal or attack). For all the records that end with the “normal”, the useful patterns will be extracted by using feature selection algorithm and saved. In our research, the normal data are collected from the kddcup.data.gz [44].

4.3.2 Detector Generation Step

In this step, all the detectors for the AIS will be generated and saved in a file. The core of this step is the matching rule that used in the negative selection algorithm. As mentioned in Chapter 3, several algorithms have been proposed by different researchers. In this experiment, a normalized Mahhatan distance is used for its simplicity. It is defined as following.

$$D(A,B) = \frac{1}{L}\sum_{i=1}\left|\frac{a(i)-b(i)}{r(i)}\right| \tag{4-7}$$

Where A = {a(1), a(2) … a(L)} , B={b(1), b(2), … b(L) } are the two patterns to be considered. R= {r(1), r(2), … r(L)}, r(i) is the range of the pattern space. In the detector generation step, the $a(i)$ represents for the random generated data and the $b(i)$ represents for the data from the KDD Cup 99 data set.

In the negative selection process, a threshold should be defined. Let $X \in Self$, Y is a pattern generated randomly, if $D(X,Y) < threshold$, then A and B are considered matching and B will be rejected. As both the attack detection rate and false alarm rate are the measurement of the system performance, the detection threshold is chosen in such a way that both rates are kept in reasonable levels. The threshold in our experiment is 0.4. The pseudocode of the negative selection process is showed as follows:

Given a shape-space Σ^L, the self set S and non-self set N. $\Sigma^L = S \cup N$ and $\Phi = S \cap N$.

```
input: S = set of self elements

output: D = set of detectors

begin

   1)  Form a self set S in shape-space Σ^L
   2)  Randomly generate a L-dimension element, and compare this element with each of
       the element in S.
   3)  If the randomly generated element fails to match any element in S, keep this
       element in set D.
   4)  If the randomly generated element matches any element in S, back to step 1.

end
```

4.3.3 Abnormal Detecting Step

The abnormal detecting part is the most important part in the intrusion detection system. And the core of this part is the matching algorithm that defines whether the input is a normal behavior or an attack. In this step, the testing data will be compared with all the detectors generated in Section 4.3.2 under certain matching rules. If the affinity between the testing data and the detectors is smaller than a threshold, which means the distance between the testing data and detector is really small. The testing data will consider as abnormal.

Let $p(i)$ is the i^{th} pattern chosen by the feature selection algorithm and $P = (p_1, p_2, \ldots p_6)$. $d(i)$ is the pattern i^{th} pattern from each record in detector set and

$D = (d_1, d_2, \ldots d_6)$. $a(i)$ is the weight coefficient of each pattern. So, the weight based Affinity definition should be

$$Affinity = \frac{1}{L}\sum_{i=1}^{6}\left|\frac{p(i)-d(i)}{range}\right| \quad (4\text{-}8)$$

The pseudocode of the detection process is shown as follow:

Given a shape-space Σ^L, the testing data set T and detector set D. The normal data in the testing data set N and Abnormal data in the testing data set A. $T = A \cup N$ and $\Phi = A \cap N$.

```
input:  T = set of testing data

output: N = set of normal data in the testing set;

        A = set of abnormal data in the testing set

begin

   1)  Form a testing data set T and a detector set D in shape-space Σ^L.
   2)  Compare the testing data with all the elements in the detector set D under
       certain decision-making algorithm.
   3)  If the testing data matches any element in the detector set D, this testing data
       will be put in set A. Otherwise, this testing data will be put in set N.

end
```

4.4 System performance of the EAI

In our experiment, the rough set algorithm (RSA) and LGP based EAI are tested. The RSA can provide a minimal subset of attributes that contains the same capability of objects classification as the whole set of attributes. It can dramatically decrease the complexity of the EAI and maintain the system performance. The LGP algorithm is tested because of its simplicity. It is fast in detecting real-time intrusions as mentioned in Section 4.2.1.2.

4.4.1 EAI using RSA

For the proposed EAI using RSA, the training data is from kddcup.data.gz [44] and the testing data can be found in corrected.gz [44]. The training data contains 743 megabytes and the testing data contains about 300,000 records. The performance of the EAI using rough set theory is shown in the following table.

Table 4-4 System Performance of EAI using RSA

TP number	FN number	FP number	TN number	TP rate	TN rate
192121	47116	17	60746	80.31%	99.97%

Comparing the Table 4-4 with Table 4-2, the EAI using rough set shows better false alarm rate, but with lower attack detection rate.

4.4.2 EAI using LGP algorithm

Similarly, the proposed EAI using LGP uses the same training data set and testing data set as the one uses rough set theory. The six parameters chosen by LGP from the KDD Cup 99 data set are service, src_byte, logged_in, rerror_rate, srv_diff_host_rate and dst_host_diff_srv_rate. The system performance of the EAI using LGP is shown in Table 4-5.

Table 4-5 System Performance of EAI using LGP

TP number	FN number	FP number	TN number	TP rate	TN rate
238880	357	9209	51554	99.85%	84.84%

Comparing the system performance of the EAI using LGP with the normal IDS using LGP shown in Table 4-3, by using the EAI, the attack detection rate increases dramatically from 89.95% to 99.85%, while the normal detection rate decreases from 94.16% to 84.84%.

4.5 Conclusion

As the performance shown in Table 4-4 and Table 4-5, it can be concluded that by introducing the AIS into the IDS, the system performance will be improved to some extent. It can either increase the TP rate or the TN rate. The problem is that by improving

one part, it will sacrifice the other part in the meantime. The major problem should be that the parameters chosen by the feature selection algorithms do not suit well for the EAI. It is assumed that all the parameters chosen by the feature selection algorithm have the same contributions to the performance of the AIS based IDS. But the reality might be that different parameters have different contributions on the system output. So, in Chapter 5, a weighted based feature selection is proposed for EAI in order to improve the IDS performance.

5 Weighted Feature Selection for EAI

The experiment presented in Chapter 4 shows that, the system performance of the EAI is not good enough by using the normal feature selection. Different parameters used in EAI might have different contributions in detecting the abnormal behavior. So, in this Chapter, a weight based feature selection for EAI is proposed and tested.

5.1 Methodology

In our experiment, an improved weight based feature selection is introduced in the abnormal detection process. Comparing with the traditional feature selection algorithm, a weight coefficient will be applied for each pattern selected by the feature selection algorithm. The algorithm is shown as follows.

Let $p(i)$ is the i^{th} pattern chosen by the feature selection algorithm and $P = (p_1, p_2, \dots p_6)$. $d(i)$ is the pattern i^{th} pattern from each record in detector set and $D = (d_1, d_2, \dots d_6)$. $a(i)$ is the weight coefficient of each pattern. So, the weight based feather selection decision-making process should be

$$Affinity = \frac{1}{L}\sum_{i=1}^{6} a(i) * \left|\frac{p(i)-d(i)}{range}\right| \qquad (5-1)$$

The weight coefficient $a(i)$ is unknown in our experiment. So the first thing is to find out a suitable weight coefficient for each parameter in different feature selection algorithms. For each feature chosen by rough set or LGP, a weight coefficient $a(i)$ will be tested. The more contribution the parameter did for the performance of IDS, the larger the weight coefficient will be. From the normalized point of view, for both rough set and LGP, the total weight of the six parameters will fix at 36, i.e.

$$\sum_{i=1}^{6} a_i = 36 \qquad (5-2)$$

The exhaustive method is used in testing the weight coefficient. It is assumed that initially, each parameter has a weight of '6' for all the six parameters chosen based on each feature selection algorithm. Then, one parameter will change by the step size of 1 and the other five will change by the step size 0.2, which keep the total weight of the equation 36 unchanged. After one parameter falls to zero, one loop will be finished.

5.1.1 Weight distribution for RSA feature set

This section presents the testing and selection of weight coefficient of the RSA feature set in our proposed EAI. The testing data can be found in corrected.gz [44]. It contains 300,000 records, and about 45 million bytes. Among those 300,000 records, 239237 records are attack patterns and 60763 records are normal patterns.

The six parameters chosen by the Rough Set theory are service, flag, src_byte, srv_count, dst_host_count, and dst_host_srv_rerror_rate (mentioned in Section 4.2.1.1). And the attack and normal detection quantities for each single changed parameter can be found in Figure 5-1 to Figure 5-6. In each figure, the horizontal axis represents for the weight of the single changed parameter, the left vertical axis represents for the number of attack detection (NoAD), and the right vertical axis represents for the number of normal confirmation (NoNC).

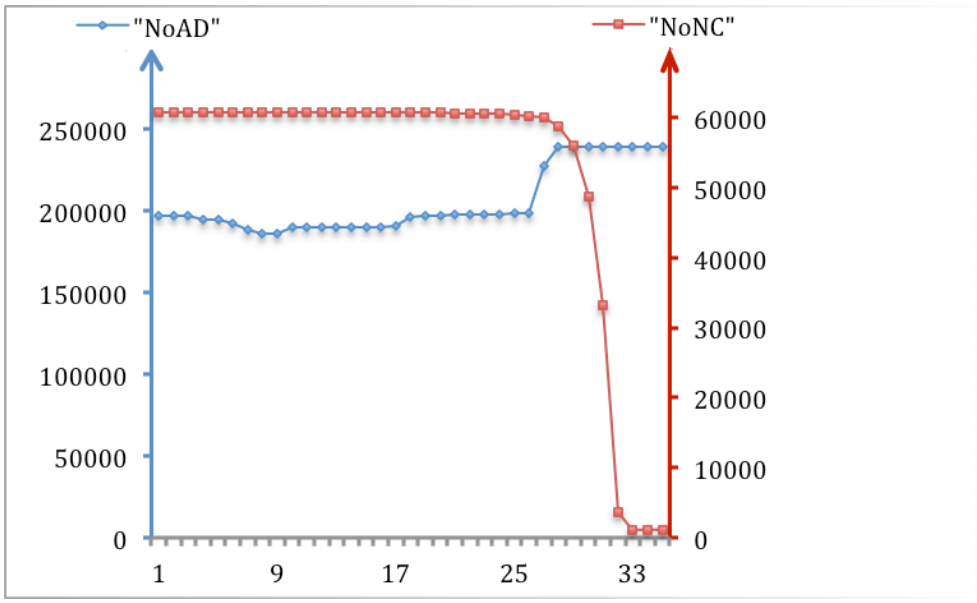

Figure 5-1 Detection results with various "service" parameter weight

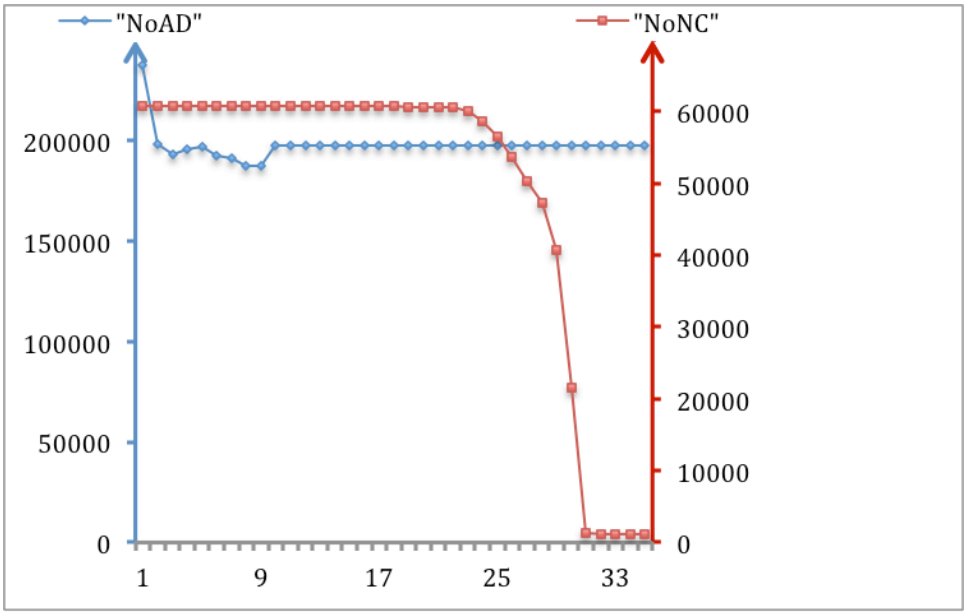

Figure 5-2 Detection results with various “flag " parameter weight

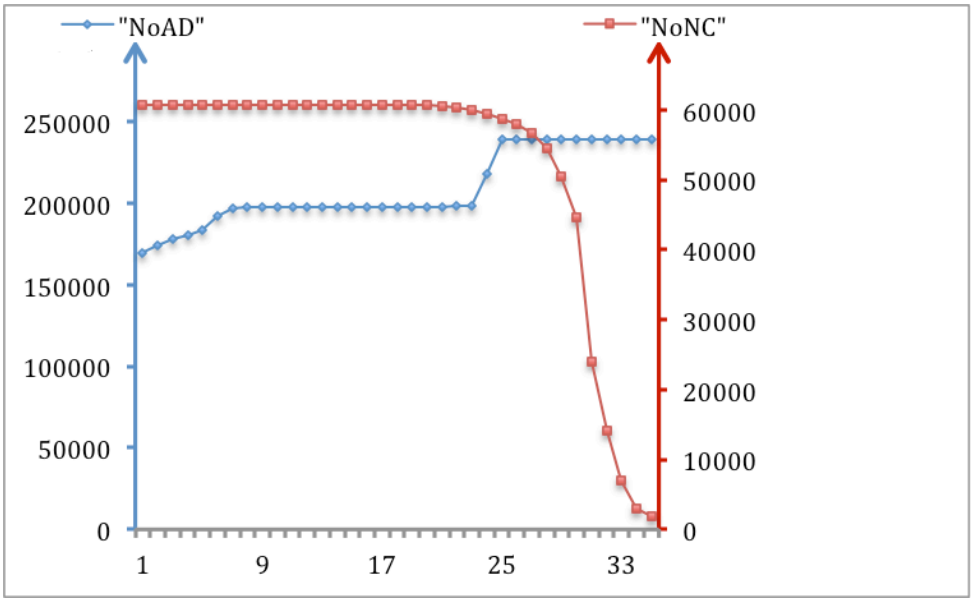

Figure 5-3 Detection results with various "src_byte" parameter weight

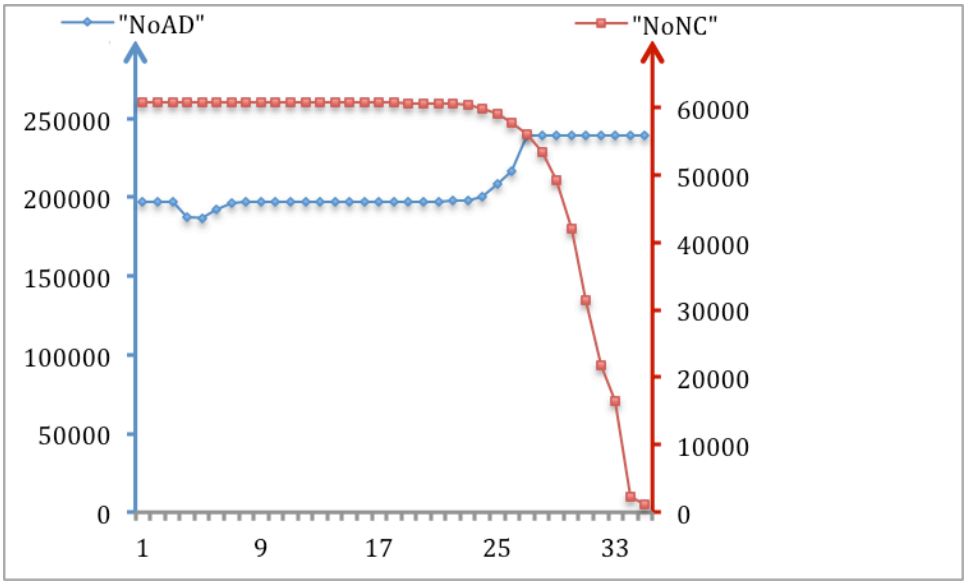

Figure 5-4 Detection results with various "srv_count" parameter weight

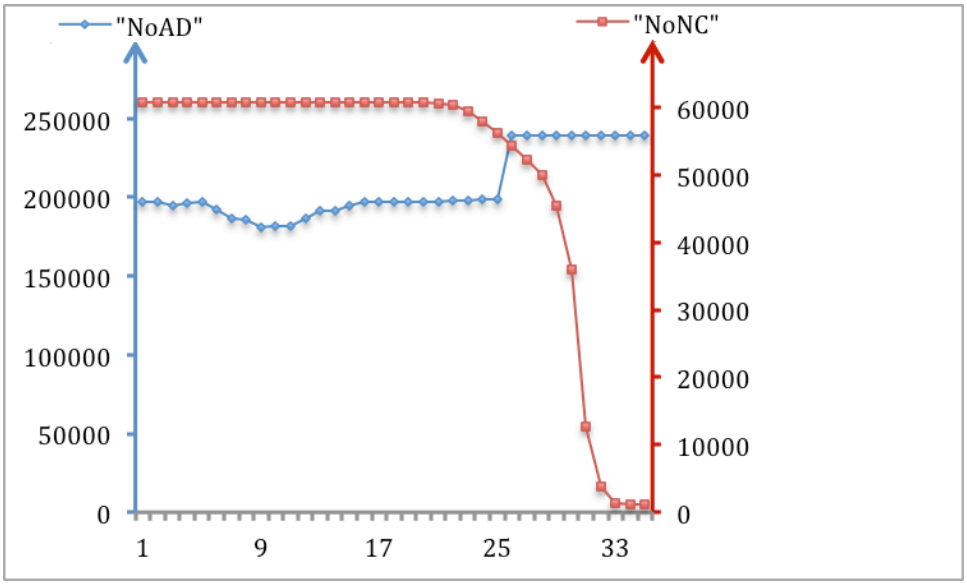

Figure 5-5 Detection results with various "dst_host_count" parameter weight

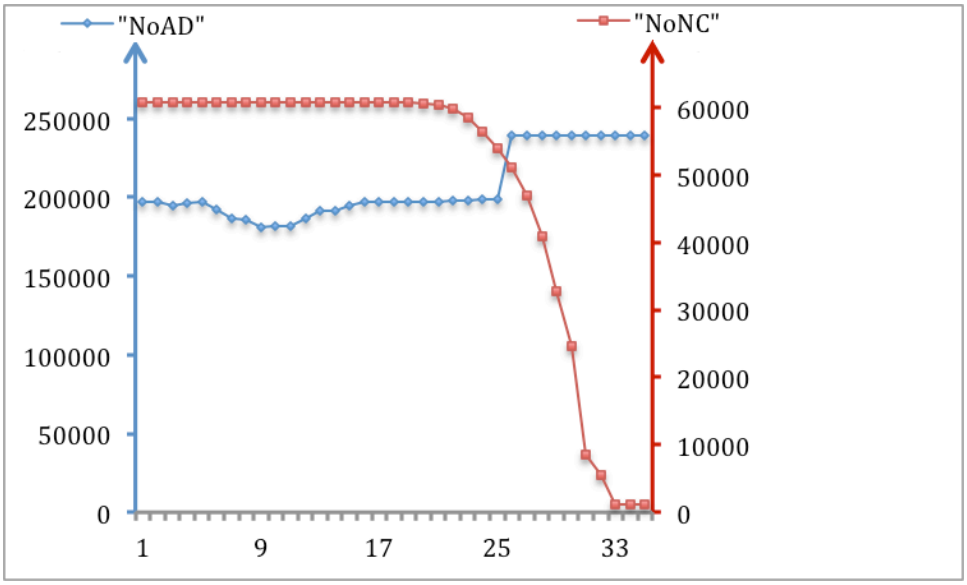

Figure 5-6 Detection results with various "dst_host_srv_rerror_rate" parameter weight

As shown in Figure 5-1 to Figure 5-6, by changing the weight of each single parameter and keep the others the same, the attack and normal detection rates vary accordingly. The table 5-1 shows the attack detection range. As shown in table 5-1, for the parameter service, src_bytes and dst_host_count, the attack detection rate change more obviously with weight adjustment. The other three parameters can be considered to have low contributions to the IDS system performance and all these parameters can have a low and fixed weight coefficient in the following test.

Table 5-1 Attack detection range for each single changed parameter in RSA

Testing parameter	service	flag	Src_bytes	Src_count	dst_host_ count	Dst_host_srv_ rerror_rate
Attack Detection Range	185843 to 239237	187200 to 237280	169610 to 239237	186953 to 239237	181493 to 239237	196076 to 239237

Now the weight coefficient combination of the other three parameters needs to be tested. The exhaustive method is used to find the best weight coefficient combinations for the rough set theory. The system performance is shown in Figure 5-7.

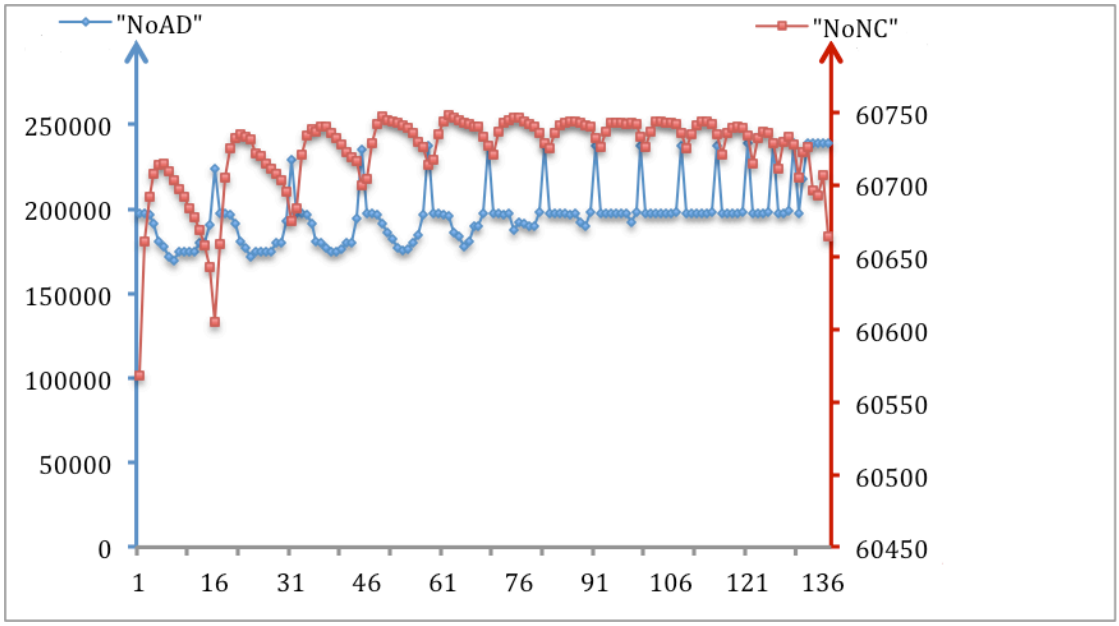

Figure 5-7 IDS performance with different weight combinations for parameters service, src_bytes and dst_host_count from RSA feature set

In Figure 5-7, the horizontal axis represents the numbered experiments with different weight coefficient combinations of the three chosen parameters. The Experiment numbers and the weight coefficients variation are mapped in Table 5-2.

Table 5-2 Value of the weight coefficient combination for EAI with weighted RSA

Experiment Number	service	src_bytes	dst_host_count	Group Number	service	src_bytes	dst_host_count	Group Number	service	src_bytes	dst_host_count
1	1	16	1	46	1	13	4	91	10	1	8
2	2	15	1	47	2	12	4	92	1	9	8
3	3	14	1	48	3	11	4	94	3	7	8
4	4	13	1	49	4	10	4	95	4	6	8
5	5	12	1	50	5	9	4	96	5	5	8
6	6	11	1	51	6	8	4	97	6	4	8
7	7	10	1	52	7	7	4	98	7	3	8
8	8	9	1	53	8	6	4	99	8	2	8
9	9	8	1	54	9	5	4	100	9	1	8
10	10	7	1	55	10	4	4	101	1	8	9
11	11	6	1	56	11	3	4	102	2	7	9
12	12	5	1	57	12	2	4	103	3	6	9
13	13	4	1	58	13	1	5	104	4	5	9
14	14	3	1	59	1	12	5	105	5	4	9
15	15	2	1	60	2	11	5	106	6	3	9
16	16	1	1	61	3	10	5	107	7	2	9
17	1	15	2	62	4	9	5	108	8	1	9
18	2	14	2	63	5	8	5	109	1	7	10
19	3	13	2	64	6	7	5	110	2	6	10

20	4	12	2	65	7	6	5	111	3	5	10
21	5	11	2	66	8	5	5	112	4	4	10
22	6	10	2	67	9	4	5	113	5	3	10
23	7	9	2	68	10	3	5	114	6	2	10
24	8	8	2	69	11	2	5	115	7	1	10
25	9	7	2	70	12	1	6	116	1	6	11
26	10	6	2	71	1	11	6	117	2	5	11
27	11	5	2	72	2	10	6	118	3	4	11
28	12	4	2	73	3	9	6	119	4	3	11
29	13	3	2	74	4	8	6	120	5	2	11
30	14	2	2	75	5	7	6	121	6	1	11
31	15	1	2	76	6	6	6	122	1	5	12
32	1	14	3	77	7	5	6	123	2	4	12
33	2	13	3	78	8	4	6	124	3	3	12
34	3	12	3	79	9	3	6	125	4	2	12
35	4	11	3	80	10	2	6	126	5	1	12
36	5	10	3	81	11	1	7	127	1	4	13
37	6	9	3	82	1	10	7	128	2	4	13
38	7	8	3	83	2	9	7	129	3	4	13
39	8	7	3	84	3	8	7	130	4	4	13
40	9	6	3	85	4	7	7	131	1	3	14
41	10	5	3	86	5	6	7	132	2	2	14
42	11	4	3	87	6	5	7	133	3	1	14
43	12	3	3	88	7	4	7	134	1	2	15
44	13	2	3	89	8	3	7	135	2	1	15
45	14	1	3	90	9	2	7	136	1	1	16

As shown in Figure 5-7, with the dynamic change of the weight coefficients, the attack and normal detection rates show an inverse correlation. A higher attack detection rate will correspond to a lower normal detection rate. The trade-offs between attack and normal detection accuracy need to be made. The purpose of the proposed EAI is to achieve a high true positive rate, and keep a relatively low false alarm rate. According to Figure 5-7, the TN rate for the proposed rough set based IDS keeps in a high level (around 99%). And the TP rate of the IDS ranges from about 70% to nearly 99%. So, the focus of our proposed IDS is to find reasonable good attack detection accuracy. The acceptable system performance of those different coefficient combinations is the TP rate up to 99.98% and the TN rate up to 99.94%. The weight coefficient combination corresponded to that performance is that 3 for service, 6 for flag, 1 for src_bytes, 6 for src_count, 14 for dst_host_count and 6 for Dst_host_srv_rerror_rate.

The comparisons of the system performance between EAI with weighted RSA and normal (or equal weighted) rough set theory is shown in Table 5-3.

Table 5-3 EAI performance of weight based RSA Feature set and normal RSA Feature Set

Type	Attack Detection Rate	Normal Detection Rate
Weight based rough set	99.98%	99.94%
Normal rough set	80.31%	99.97%

According to Table 5-3, by introducing the weighted based scheme, the performance of the proposed EAI improves significantly in detecting abnormal behaviors, nearly 19%. Both the normal detection accuracy and attack detection rate can be above 99.9 percent.

5.1.2 Weight distribution for LGP Feature Set

Similar as Section 5.1.1, in this section, the weight coefficient of each parameter for the LGP feature set, which, has six parameters, i.e. service, src_byte, logged_in, rerror_rate, srv_diff_host_rate and dst_host_diff_srv_rate. The numbers of correct attack and normal detections for varying weights on each parameter can be found in Figure 5-8 to Figure 5-13. They are same as Figure 5-1 to Figure 5-6, the horizontal axis represents for the weight associated with the single testing parameter, the left vertical axis represents for the number of attack detection (NoAD), and the right vertical axis represents for the number of normal confirmation (NoNC).

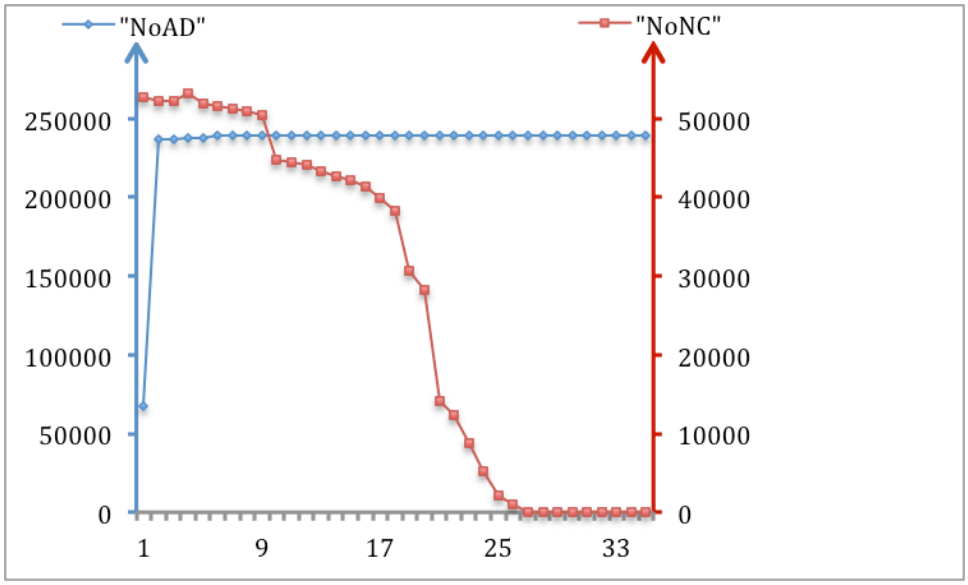

Figure 5-8 Detection results with various “service” parameter weight

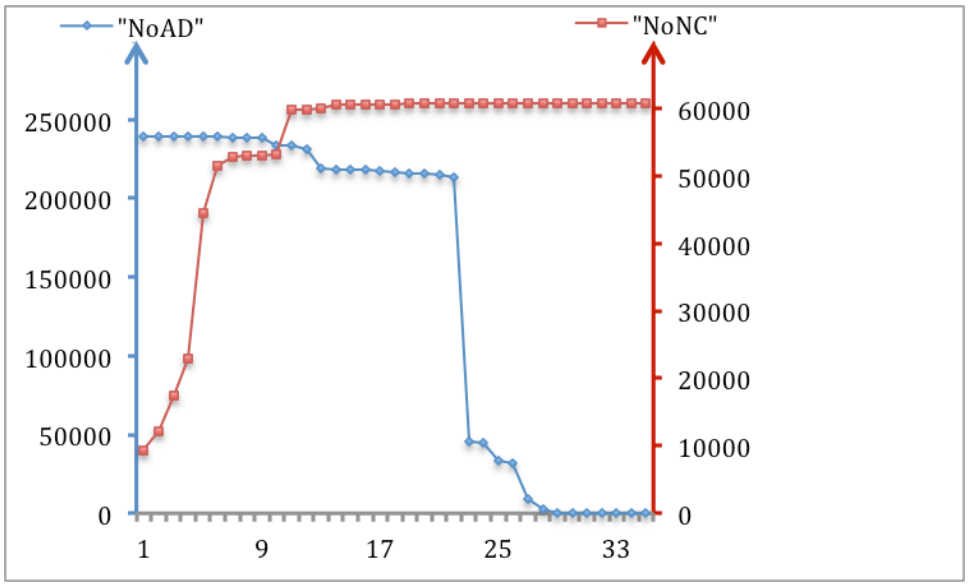

Figure 5-9 Detection results with various "src_byte" parameter weight

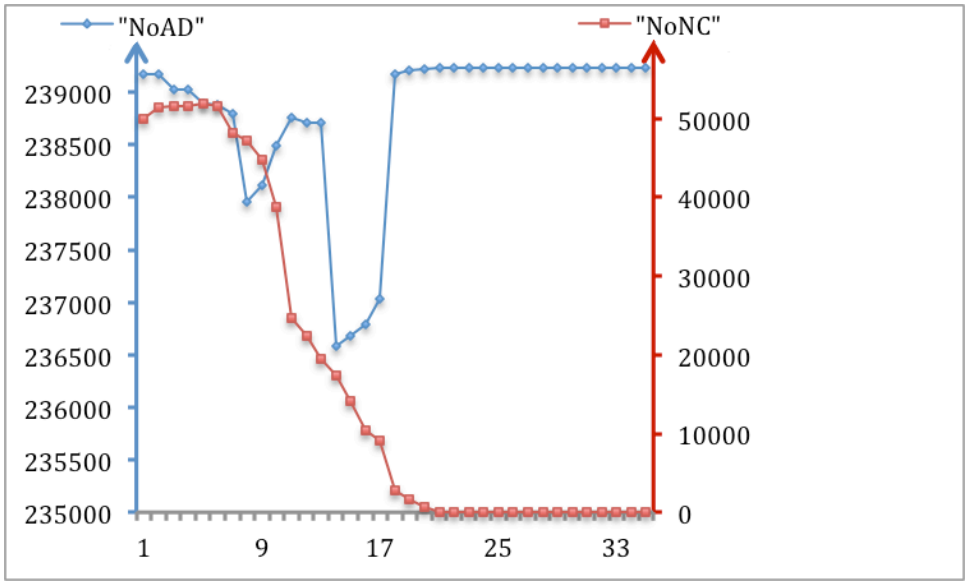

Figure 5-10 Detection results with various "logged_in" parameter weight

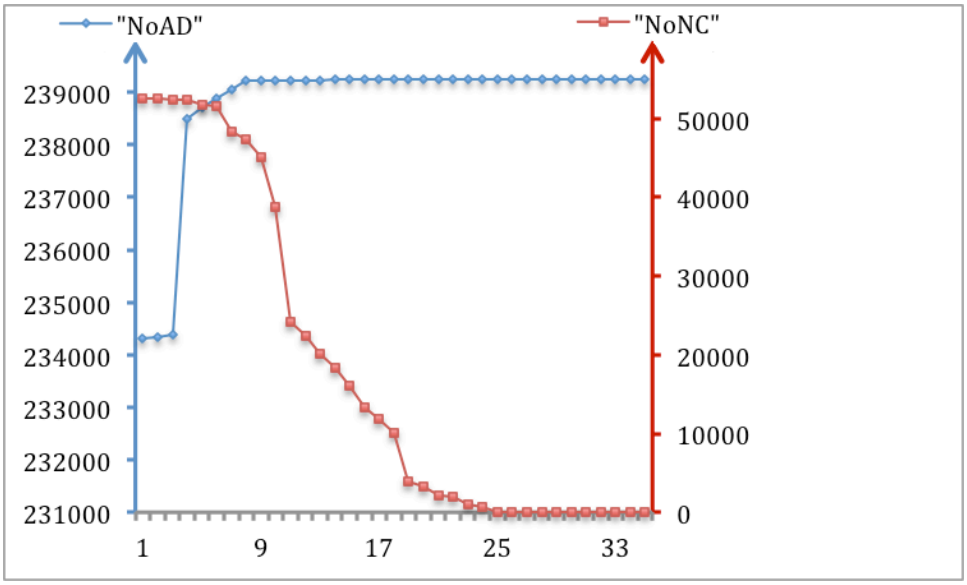

Figure 5-11 Detection results with various "rerror_rate" parameter weight

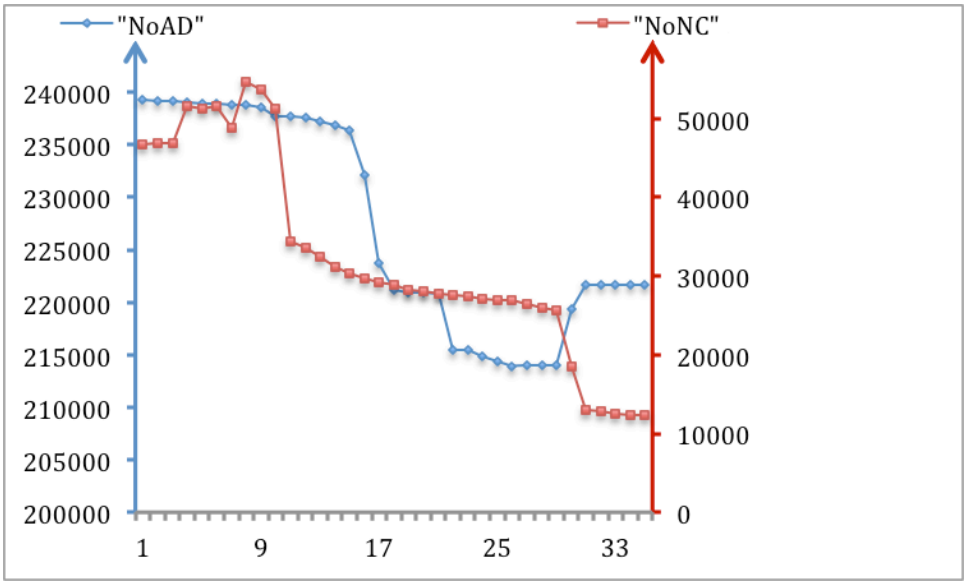

Figure 5-12 Detection results with various "srv_diff_host_rate" parameter weight

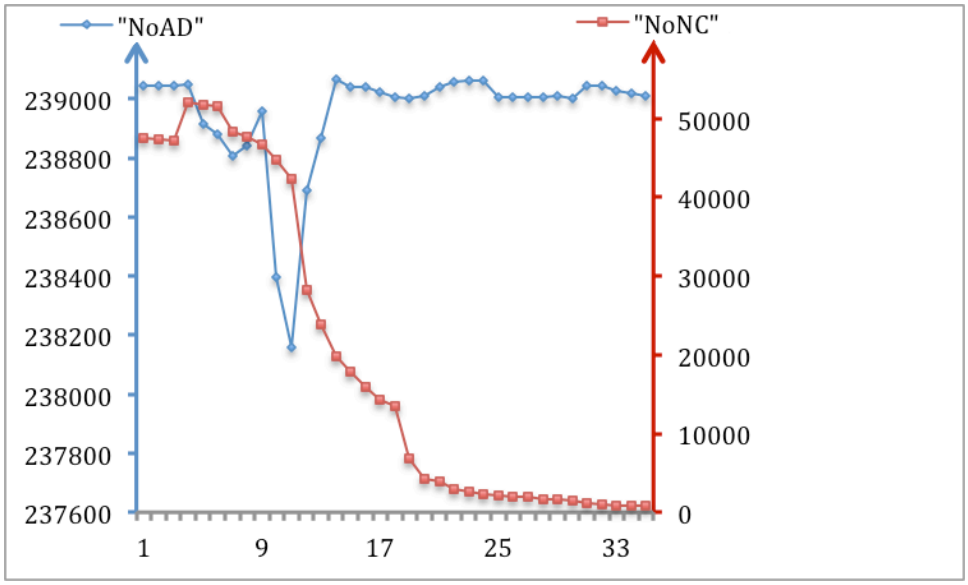

Figure 5-13 Detection results with various "dst_host_diff_srv_rate" parameter weight

Compared with Figure 5-1 ~ Figure 5-6, Figure 5-8 ~ Figure 5-13, have shown that even some features appears in both RSA feature set and LGP feature set, e.g. service and src_byte, the contributions to the IDS performance are quite different. The attack detection range for each testing parameter from LGP feature set is shown in Table 5-4.

Table 5-4 Attack detection range for each single changed parameter in LGP

Testing parameter	service	src_byte	logged_in	rerror_rate	srv_diff_host _rate	dst_host_diff_srv _rate
Attack Detection Range	67828 to 239237	0 to 239235	236583 to 239237	234309 to 239237	213944 to 239237	238159 to 239066

Table 5-3 shows that three parameters, service, src_byte, and srv_diff_host_rate have a higher influence (measured by the range of attack detection quantity) on the system output than the others. Similarly to the rough set algorithm, the exhaustive method is used to find the best weight coefficient combinations of these three parameters. The performance of the EAI with weighted LGP intrusion detection system is shown in Figure 5-14.

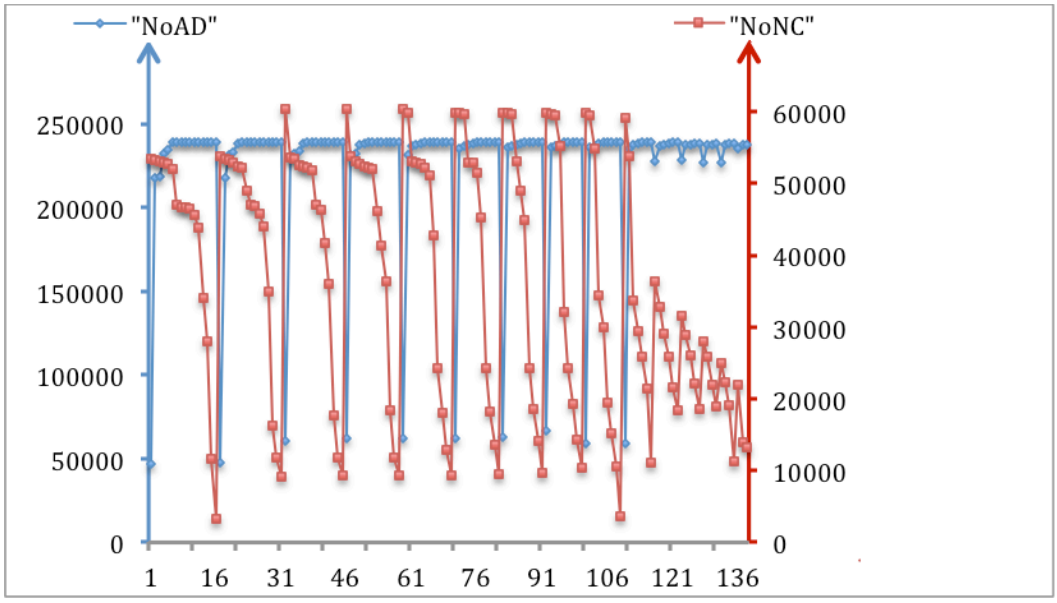

Figure 5-14 IDS performance with different weight combinations for parameters service, src_byte, and srv_diff_host_rate from LGP feature set

The value of the weight coefficient combination for the EAI with weighted LGP is shown in Table 5-5.

Table 5-5 Value of the weight coefficient combination for EAI with weighted LGP

Group Number	service	src_bytes	srv_diff_host_rate	Group Number	service	src_bytes	srv_diff_host_rate	Group Number	service	src_bytes	srv_diff_host_rate
1	1	16	1	46	1	13	4	91	10	1	8
2	2	15	1	47	2	12	4	92	1	9	8
3	3	14	1	48	3	11	4	94	3	7	8
4	4	13	1	49	4	10	4	95	4	6	8
5	5	12	1	50	5	9	4	96	5	5	8
6	6	11	1	51	6	8	4	97	6	4	8
7	7	10	1	52	7	7	4	98	7	3	8
8	8	9	1	53	8	6	4	99	8	2	8
9	9	8	1	54	9	5	4	100	9	1	8
10	10	7	1	55	10	4	4	101	1	8	9
11	11	6	1	56	11	3	4	102	2	7	9
12	12	5	1	57	12	2	4	103	3	6	9
13	13	4	1	58	13	1	5	104	4	5	9
14	14	3	1	59	1	12	5	105	5	4	9
15	15	2	1	60	2	11	5	106	6	3	9
16	16	1	1	61	3	10	5	107	7	2	9
17	1	15	2	62	4	9	5	108	8	1	9
18	2	14	2	63	5	8	5	109	1	7	10
19	3	13	2	64	6	7	5	110	2	6	10
20	4	12	2	65	7	6	5	111	3	5	10
21	5	11	2	66	8	5	5	112	4	4	10
22	6	10	2	67	9	4	5	113	5	3	10
23	7	9	2	68	10	3	5	114	6	2	10
24	8	8	2	69	11	2	5	115	7	1	10
25	9	7	2	70	12	1	6	116	1	6	11
26	10	6	2	71	1	11	6	117	2	5	11
27	11	5	2	72	2	10	6	118	3	4	11
28	12	4	2	73	3	9	6	119	4	3	11
29	13	3	2	74	4	8	6	120	5	2	11
30	14	2	2	75	5	7	6	121	6	1	11
31	15	1	2	76	6	6	6	122	1	5	12
32	1	14	3	77	7	5	6	123	2	4	12
33	2	13	3	78	8	4	6	124	3	3	12
34	3	12	3	79	9	3	6	125	4	2	12
35	4	11	3	80	10	2	6	126	5	1	12
36	5	10	3	81	11	1	7	127	1	4	13
37	6	9	3	82	1	10	7	128	2	4	13
38	7	8	3	83	2	9	7	129	3	4	13
39	8	7	3	84	3	8	7	130	4	4	13
40	9	6	3	85	4	7	7	131	1	3	14
41	10	5	3	86	5	6	7	132	2	2	14
42	11	4	3	87	6	5	7	133	3	1	14
43	12	3	3	88	7	4	7	134	1	2	15
44	13	2	3	89	8	3	7	135	2	1	15
45	14	1	3	90	9	2	7	136	1	1	16

Figure 5-14 and Table 5-5 show that the attack detection rate and the normal detection rate are also reversely correlated. Compared with the performance of EAI with weighted RSA, the LGP version has higher fluctuation on the two detection rates. The TP rate ranges from 20% to 100%, while the TN rate is ranging from 99% to 15%. Acceptable system performance should have relatively high normal confirmation rate and attack detection rate. As shown in Figure 5-14, one group of weight setting can achieve the following encouraging result, TP rate: 98.91% and the TN rate: 98.24%. The weight coefficient distribution is as following: 3 for service, 8 for src_byte, 6 for logged_in, 6 for rerror_rate, 7 for srv_diff_host_rate, and 6 for dst_host_diff_srv_rate.

The comparison of the EAI performance between weight based LGP and traditional LGP algorithm is shown in Table 5-6.

Table 5-6 EAI performance of weight based LGP Feature set and normal LGP Feature Set

Type	Attack Detection Rate	Normal Detection Rate
Weight based LGP	98.91%	98.24%
Normal LGP	99.85%	84.84%

The Table 5-6 shows that, by introducing the weigh coefficient to the feature selected by the LGP, the false alarm of the EAI decreases dramatically, about 14%. The performance of the EAI with weighted LGP is not as good as the one using RSA mainly because the RSA features presents better for the whole data set. But still, the LGP has its advantage because its feature extracting process is quicker and fit for real-time intrusion detection as discussed in Chapter 4.

5.2 Discussion

The IDS performances shown in Section 5.1 indicate that the introduction of the weigh coefficients into the feature selection algorithm can improve the performance of the system. The comparison of the performance of EAI with weighted features and other IDSs is shown in Table 5-7

Table 5-7 Comparisons of EAI with weighted features and other IDS systems

Type	Attack Detection Rate	Normal Detection Rate
EAI with weighted LGP	98.91%	98.24%
EAI with weighted RSA	99.98%	99.94%
MARS[45]	99.925%	84.9%
SVDF[45]	99.928%	80.83%
DT[42]	92.088%	99.998%
HGMM[52]	98.775%	88.14%

As shown in Table 5-7, the EAI with weighted RSA performs best comparing with other intrusion detection systems. By changing the weight coefficient of different parameters, the IDS system designers have greater flexibility in making tradeoffs between the TP rate and TN rate. The difficulty of the proposed scheme is the selection of the weight coefficients. The use of the exhaustive method in chosen the weight coefficients requires a lot of computing time. With all the system output from the exhaustive method, a trade-off between attack detection rate and false alarm rate needs to be made. A good intrusion detection system should have a high abnormal detection rate and relatively low false alarm rate. The advantage of the proposed weighted EAI is that it can suit different network circumstances by changing the weight coefficients.

6 Conclusion

This research targets exploration of network intrusion detection (ID), an important sub area in network security. The objective is to further improve existing ID technologies' detection accuracy and system complexity. The capability of detecting new attacks or virus is also required for the targeted intrusion detection system (IDS). After the most popular intrusion detection techniques are reviewed, Artificial Immune System (AIS), a bio inspired computing paradigm is selected for our research in ID. An efficient AIS based IDS (EAI) scheme is proposed in this thesis. The scheme relies on the detectors to detect the abnormal behaviour of the network. The detectors are generated by using the negative selection AIS algorithm. The nature of this scheme enables it to detect new attacks. Feature selection is to limit the features used in the scheme to be a small group which are more reprentitive from abnormal detection point of view. The small group means a low computing complexity. This is an issue in real time applications. Two feature selection algorithms, rough set algorithm (RSA) and linear genetic programming (LPG), have been studied, compared and tested in the scheme. Rough set algorithm can provide a minimal subset of attributes that contains the same capability of objects classification as the whole set of attributes. LPG has its own advantages in process speed. That is why both algorithms are studied in this research. EAI with either RSA or LPG feature set has shown some encouraging results. However, as all selected features in both algorithms are treated equally in the EAI scheme, the detection performance should be further improved. For the reason that the selected features may not have equal contribution towards the detections, a weighted EAI is

proposed in aiming to further improve the detection performance. In the weighted scheme, all the features used should be assigned with individual weight associated with its contribution to the detection. The weight allocation is the main challenge. In this thesis, a simplified exhausted method is proposed in the weight selection. The weighted EAI with both RSA and LPG feature sets have been tested. The results show that the attack detection rate of the rough set based IDS can increase by 19.67% and the false alarm rate of the LGP based IDS can decrease by 13.4%. With both RSA and LPG, the weight EAI has made clear improvement in both attack detection rate and the normal confirmation rate. The best result obtained is from weighted EAI with RSA feature set. The TP rate is 99.98% and TN rate is 99.94%. The testing data is the popular KDD Cup 99 data set [44]. In summary, the work conducted in this research has shown that artificial immune system has a big role to play in the intrusion detection area even there are some works yet to be done in real implementation.

KDD CUP 99 is the most popular data set used in the intrusion detection research field, but it is more than eleven years old, may not reflect some new features and attacks of today's networks. With fast evolution of networks and the Internet, new updated data set recognized by the research community is needed. This is challenging work as the generation of the data set requires resources and proper design. Hopefully, Lincoln Lab will update their data set soon. Otherwise alternative test data needs to be obtained.

Due to the time limit, the adaptiveness of AIS based IDS has not been explored. In our scheme, once the detectors are generated, they are supposed to be fixed in the live detection. However, some detectors may not detect anything for long time. Inclusion of these detectors in the searching process of the scheme may not add much value to the system. These inactive detectors can be put into a sleep mode for the scheme to have efficient computing. Also, a mechanism that can generate new detectors could be added to adapt to the network changes. These can be the future works in the area.

7 References:

[1] Hans-Peter Kriegel, Peer Kröger, Arthur Zimek (2009). "Outlier Detection Techniques (Tutorial)". 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2009) (Bangkok, Thailand).

[2] Cristianini, N., Taylor, S.J, "An Introduction to Support Vector Machines." Cambridge University Press (2000)

[3] Friedman, J.H, "Multivariate Adaptive Regression Splines." Annals of Statistics; Vol.19. (1991) 1-141

[4] Banzhaf, W., Nordin, P., Keller, E.R., Francone, F.D, "Genetic Programming: An Introduction on the Automatic Evolution of Computer Programs and its Applications." Morgan Kaufmann Publishers, Inc (1998)

[5] J.D. Farmer, N. Packard and A. Perelson, (1986) "The immune system, adaptation and machine learning", Physica D, vol. 2, pp. 187–204

[6] H. Bersini, F.J. Varela, Hints for adaptive problem solving gleaned from immune networks. Parallel Problem Solving from Nature, First Workshop PPSW 1, Dortmund, FRG, October, 1990.

[7] Leandro N. de Castro and Jonathan Timmis, "Artificial Immune Systems: A New Computational Intelligence Approach", Springer, 2002.

[8] S. A. Hofmeyr and S. Forrest, "Immunity by design: An artificial immune system," in Proceedings of the Genetic and Evolutionary Computation Conference. San Mateo, CA: Morgan Kaufmann, July 1999, pp.1289–1296.

[9] D. Dasgupta. Immunity-based intrusion detection systems: A general framework. presented at 22nd Nat. Information Systems Security Conf.. [Online]. Available: http://csrc.nist.gov/nissc/1999/proceedings/papers/ p11.pdf

[10] J.D. Farmer, N.H. Packard, A.S. Perelson, The immune system, adaptation, and machine learning, Physica D 22 (1986) 187–204.

[11] A.S. Perelson, Immune network theory, Immunological Reviews 110 (1989) 5–36.

[12] F. Varela, A. Coutinho, B. Dupire, N. Vaz, Cognitive networks: Immune, neural and otherwise, Theoretical Immunology 2 (1988) 359–375.

[13] Tizard, I. R., 1995, Immunology: Introduction, 4th Ed, Saunders College Publishing.

[14] Leandro N. de Castro and Jonathan Timmis, "Artificial Immune Systems: A New Computational Intelligence Approach", Springer, 2002.

[15] Kim, J. and Bentley, P., 1999a, The Human Immune System and Network Intrusion Detection, 7th European Conference on Intelligent Techniques and Soft Computing (EUFIT '99), Aachen, Germany (to appear).

[16] Kim, J and Bentley, PJ (1999) Negative selection and niching by an artificial immune system for network intrusion detection. In: (Proceedings) Proc. Late-Breaking Papers at the Genetic and Evolutionary Computation Conference (GECCO'99), Orlando, Florida. (pp. 149 - 158).

[17] Forrest, S., Hofmeyr, S., and Somayaji, A., 1997, Computer Immunology, Communications of the ACM, 40(10), 88-96.

[18] Jonathan Timmis, Andrew Hone, Thomas Stibor, Edward Clark: Theoretical advances in artificial immune systems. Theor. Comput. Sci. 403(1): 11-32 (2008)

[19] J.Kelsey, J.Timmis, Immune inspired somatic contiguous hypermutation for function optimization, in: Genetic and Evolutionary Computation Conference, in: Lecture Notes in Computer Science, vol.2723, Springer, 2003, pp. 207-218.

[20] De Castro, L. N. & Von Zuben, F. J. (2000). The Clonal Selection Algorithm with Engineering Applications, Proceedings of Genetic and Evolutionary Computation Conference, Las Vegas, Nevada, USA, July, 2000, pp. 36-37.

[21] Leandro N. de Castro and Jon Timmis(2002). An artificial immune network for multimodal function optimization. In IEEE Congress on Evolutionary Computation (CEC), pages 699–704,.

[22] N.K. Jerne, Towards a network theory of the immune system, Ann. Immunol. (Inst. Pasteur) 125C (1974) 373–389.

[23] L.N. de Castro, F.J. von Zuben, aiNet: An artificial immune network for data analysis, in: Hussein A. Abbass, Ruhul A. Sarker, Charles S. Newton (Eds.), Data Mining: A Heuristic Approach, Idea Group Publishing, 2001, pp. 231–259 (Chapter 12).

[24] Forrest, S.; Perelson, A.S.; Allen, L.; Cherukuri, R. (1994). "Self-nonself discrimination in a computer". Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy. Los Alamitos, CA. pp. 202–212.

[25] S Forrest, S Hofmeyr, and A Somayaji. Computer Immunology. Communications of the ACM, 40(10):88–96, 1997.

[26] J Kim and P Bentley. Immune Memory in the Dynamic Clonal Selection Algorithm. In J Timmis and P Bentley, editors, Proceedings of the First International Conference on Artificial Immune Systems ICARIS, pages 59–67, 2002.

[27] J. Balthrop, F. Esponda, S. Forrest, and M. Glickman. Coverage and generalization in an artificial immune system. In GECCO 2002: Proc. of the Genetic and Evolutionary Computation Conf., pages 3–10, 2002.

[28] D' haeseleer , P , 1997, A Distributed Approach to Anomaly Detection, *ACM Transactions on Information System Security*.

[29] R. Heady, G. Luger, A. Maccabe, and M. Servilla. The archi- tecture of a network level intrusion detection system. Technical report, Computer Science Department, University of New Mexico, August 1990.

[30] T. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, P. Neumann, H. Javitz, A. Valdes, and T. Garvey. A real-time intrusion detection expert system (IDES) - final technical report. Technical report, Computer Science Lab- oratory, SRI International, Menlo Park, California, February 1992.

[31] V. Paxson. Bro: A system for detecting network intruders in real- time. In Proceedings of the 7th USENIX Security Symposium, San Antonio, TX, 1998.

[32] T. Lunt. Detecting intruders in computer systems. In Proceedings of the 1993 Conference on Auditing and Computer Technology, 1993.

[33] Patcha, A. and Park, J.-M. 2007. An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks 51, 12, 3448-3470.

[34] Anazida Zainal, Mohd Aizaini Maarof and Siti Mariyam Shamsuddin, "Feature selection using rough set in intrusion detection", Tencon 2006. 2006 IEEE Region 10 Conference, pp.1-4.

[35] Mattord, verma (2008). *Principles of Information Security*. Course Technology. pp. 290–301. ISBN 9781423901778

[36] T. Stibor, J. Timmis, and C. Eckert. A Comparative Study of Real-Valued Negative Selection to Statistical Anomaly Detection Techniques . In C. Jacob, M. Pilat, P. Bentley, and J. Timmis, editors, Proceedings of the 4th International Conference on Artificial Immune Systems, volume 3627 of LNCS, pages 262– 275. Springer, 2005.

[37] U. Aickelin and S. Cayzer. The danger theory and its application to artificial immune systems. In Jonathan Timmis and Peter J. Bentley, editors, Proceedings of the 1st International Conference on Artificial Immune Systems ICARIS, pages 141–148, University of Kent at Canterbury, September 2002. University of Kent at Canterbury Printing Unit.

[38] U Aicklen, P Bentley, S Cayzer, J Kim, and J McLeod. Danger Theory: The Link Between AIS and IDS? In LNCS 2787, pages 147–155. Springer, 2003.

[39] M. Ali Aydın, A. Halim Zaim, K. Gökhan Ceylan, "A hybrid intrusion detection system design for computer network security", Computers and Electrical Engineering 35 (2009) 517–526.

[40] Alonso-Betanzos, A., Sánchez-Maroño, N., Carballal-Fortes, F. M., Suárez-Romero, J., & Pérez-Sánchez, B. (2007). Classification of computer intrusions using functional networks. A comparative study. In Proceedings of the 15th European symposium on artificial neural networks (ESANN'07) (pp. 579–584).

[41] Shyu, M.-L., Chen, S.-C., Sarinnapakorn, K., & Chang, L.-W. (2003). A novel anomaly detection scheme based on principal component classifier. In IEEE Foundations and new directions of data mining workshop in conjunction with the third IEEE international conference on data mining (ICDM'03) (pp. 172– 179).

[42] K.Saravanan, "An Efficient Detection Mechanism for Intrusion Detection Systems Using Rule Learning Method", International Journal of Computer and Electrical Engineering, Vol. 1, No. 4, October, 2009 1793-8163

[43] R.P.Lippmann, D.J.Fried, I.Graf, J.W.Haines, K.R.Kendall, D.McClung, D.Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman, "Evaluating intrusion detection systems: The 1998 darpa off- line intrusion detection evaluation," discex, vol. 02, p. 1012, 2000.

[44] KDD99, KDD cup 1999 data. (1999). <http://kdd.ics.uci.edu/databases/kddcup99/ kddcup99.html>.

[45] A. H. Sung, and S. Mukkamala, *"The Feature Selection and Intrusion Detection Problems". Springer Verlag Lecture Notes Computer Science 3321*. 2004, pp. : 468-482.

[46] L. Zhang, G. Zhang, L. Yu, J. Zhang, and Y. Bai, "Intrusion Detection Using Rough Set Classification." *Journal of Zheijiang University Science*. 2004 5(9), pp. 1076-1086.

[47] S. Chebrolu, A. Abraham, and J. P. Thomas, "Features Deduction and Ensemble Design of Intrusion Detection Systems." *Journal of Computers and Security*, Volume 24, Issue 4, June 2005, pp. 295-307.

[48] Pawlak,Z., 1982. Rough sets. International Journal of Computer and Information Sciences, 11:341-356.

[49] R. Jensen, and S. Qiang, "Finding Rough Set Reducts with Ant Colony Optimization." *Proceedings of the 2003 UK Workshop on Computational Intelligence*, 2003, pp. 15-22.

[50] Banzhaf, W., Nordin, P., Keller, E.R., Francone, F.D.: Genetic Programming: An Introduction on the Automatic Evolution of Computer Programs and its Applications. Morgan Kaufmann Publishers, Inc (1998)

[51] Brameier, M, Banzhaf, W.: A Comparison of Linear Genetic Programming and Neural Networks in Medical Data Mining. IEEE Transactions on Evolutionary Computation; Vol. 5(1). (2001) 17-26

[52] M. Bahrololum and M. Khaleghi, "Anomaly Intrusion Detection System Using Hierarchical Gaussian Mixture Model", Journal of Computer Science (2008) Volume: 8, Issue: 8, Publisher: IEEE, Pages: 264-271

Appendix A

Network Intrusion Detection by Artificial Immune System

Junyuan Shen
RMIT University
Melbourne, Australia
junyuan.shen@student.rmit.edu.au

Jidong Wang
RMIT University
Melbourne, Australia
jidong.wang@rmit.edu.au

***Abstract*—With the increasing network attacks worldwide, intrusion detection (ID) has become a hot research topic in last decade. Technologies such as neural networks and fuzzy logic have been applied in ID. The results are varied. Intrusion detection accuracy is the main focus for intrusion detection systems (IDS). Most research activities in the area aim to improve the ID accuracy. In this paper, an artificial immune system (IMS) based network intrusion detection scheme is proposed. An optimized feature selection and parameter quantization algorithms are defined. The complexity issue is addressed in the design of the algorithms. The scheme is tested on the widely used KDD CUP 99 dataset. The result shows that the proposed scheme outperforms other schemes in detection accuracy. In our experiments, a number of feature sets have been tried and compared. Compromise between complexity and detection accuracy has been discussed in the paper.**

Keywords- Intrusion Detection, Negative selection, Artificial Immune System, KDD CUP 99

I. INTRODUCTION

With the enormous development of the computer and network technologies, the security of the network information is becoming increasingly important. New access technologies and devices have increased the possibilities of malicious attacks or service abuse by various hackers. The traditional passive defense technologies like encryptions and firewalls can not fully meet the current security requirements. Therefore, the Intrusion Detection Systems (IDSs) which serves as special purpose systems to detect attacks and misuses in the network is needed.

Generally speaking, two approaches, misuse detection and anomaly detection, can be used in computer systems and computer networks. The misuse detection is used to detect the intrusion when the behavior of the system matches with any of the intrusion signatures. And the anomaly detection, also called as outlier detection [3], is used to detect the intrusion when the given data set does not match with the established normal behavior. Modern IDSs usually combines both of these two approaches.

Various techniques have been used for building IDS, like Support Vector Machines (SVM) [17], Multivariate Adaptive Regression Splines (MARS) [18], and Linear Genetic Programming (LGP) [19], etc. Some of them give good performance in specific attack areas, while they might not detect other attacks well. In recent years, bio-inspired algorithms have been studied and applied in intrusion detection [11] aiming for better performance. Algorithms such as Genetic Algorithm (GA), Artificial Neural Networks (ANN) and Artificial Immune Systems are widely studied. Among them, AIS is a relatively new comer. Further investigation on AIS based network intrusion detection is needed. The concept of AIS was proposed in mid 1980s. Farmer, Packard and Perelson [29], Bersini and Varela's [30] work have started the area. AIS became a subject of its own in mid 90s. It has been defined by Castro and Timmis [5] as: "Adaptive systems inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving." The early work of applying AIS to IDS can be found in [11]. A multilayer AIS based IDS was proposed by Dasgupta [12] in order to provide systematic defense. These AIS based algorithms have achieved good detection results. But their computing complexity is quite high. In IDS, responding time is an important issue. The more complex the system, the more computing time and the longer responding time will be. Large parameter set in IDS can increase the detection accuracy. However, the more parameters used, the more complex the system. The trade off between the complexity and the accuracy is a challenge. Our study on AIS based IDS is to further improve its detection accuracy while keeping a low algorithm complexity.

In this paper, an AIS based intrusion detection system with some efficient feature selection algorithms is presented. The anomaly detection in the system is set up based on AIS negative selection algorithm. The feature selection algorithm is used to reduce the complexity of the system. The artificial immune system and the negative selection algorithm are introduced in Section II. The AIS based IDS is presented in Section III. Our experiment and results are illustrated in Section IV. Section V draws a conclusion and some future works are discussed.

II. ARTIFICIAL IMMUNE SYSTEM

The artificial immune system (AIS) is a branch of bio-inspired computational intelligence, and it has attacked increasing interest from the researchers after it was first proposed. Three main algorithms: negative selection, clonal selection, immune network theory compose the most popular

selection during the maturation of T cells in the thymus. It is the major algorithm of the artificial immune system. In the case of an anomaly detection domain, the algorithm prepares a set of exemplar pattern detectors trained on normal (non-anomalous) patterns that model and detect unseen or anomalous patterns [6]. The principle of the negative selection is shown in Figure 1.

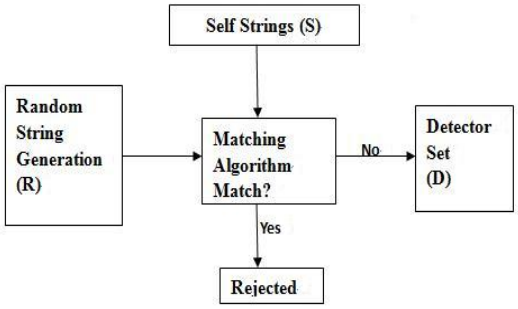

Figure 1. The Principle of Negative Selection

As shown in Figure 1, the matching algorithm is the core of the negative selection. The affinity between the Ab (Antibody) and Ag (Antigen) is decided by using the matching algorithm. Several algorithms have been proposed in this area to determine the affinity, like Eucliden algorithm, hamming distance algorithm, r-contiguous bit rule algorithm, etc. The Affinities of the Ab and Ag are related to the distance between them. The definition of the distance can be shown as follows

Let Ab=<Ab_1, Ab_2… Ab_m>, Ag=<Ag_1, Ag_2… Ag_m>,

- Eucliden $D = \sqrt{\sum_{i=1}^{m} (Ab_i - Ag_i)^2}$
- Manhattan $D = \sum_{i=1}^{m} |Ab_i - Ag_i|$
- Hamming $D = \sum_{i=1}^{m} \delta$, where $\delta = \begin{matrix} 1 & if \quad Ab_i \neq Ag_i \\ 0 & otherwise \end{matrix}$

For negative selection, if the distance between the Ab and the self string is lager then a threshold, which means it matches with the Ag, the Ab will be eliminated. Otherwise, the Ab will be kept as a detector.

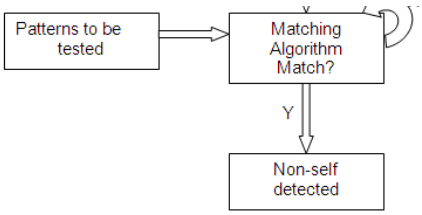

Figure 2. The detection process

After the detector set is created, it can be used for detec of non-self elements. As shown in Figure 2, for any patter be checked, it needs to be compared with all the patterns in detector set. If it is matched to any pattern in the detector s will be considered as a non-self element. AIS has been fo applications in many areas such as optimization, data analy machine learning, pattern recognition, etc and netw intrusion detection which is the focus of this paper.

III. AIS BASED IDS

Generally, network intrusion detection is based on examination of monitored network parameters. Diffe examine algorithms lead to different IDS. The general based IDS [26] [27] [28] can be divided into two parts, detector set generation and non-self detection. To form detection set, negative selection algorithm is applied. A l set of normal network parameter patterns are required. Initi the immature detectors (parameter pattern in this case) randomly generated as shown in figure 1. Then, t immature detectors are compared with the normal netw parameter patterns. If a random generated pattern matche normal pattern, the immature detector will be rejected deleted. Those which do not match any normal netw parameter patterns will be saved as mature detectors. In live detection stage, a monitored network parameter patter compared with detectors in the detector set. If it is matc with any detector, then a network intrusion is detected.

A compact and effective detector set can reduce algorithm computing complexity. For detectors which do contribute any detection in a period of time, they should removed or put to a sleeping state. Therefore, all the ma detectors will have a time_to_live (TTL) parameter. When detection is occurred, all detectors' TTLs will be deductec one except for the detector which detects the intrusion. TTLwill be reset to the maximum. When a detector reache lifetime, ie its TTL becomes zero; this detector will be bec inactive.

a) Experimental Database

The KDD Cup 99 data set, which is the most widely used data set for network-based intrusion detection, is used in our project. This data set is built based on the data captured in DARPA'98 IDS evaluation program [13].The data set contains 24 training attack types and 14 additional attack types in the test data only. These attacks fall into four main categories:

1. **Denial of service (DOS):** In this type of attack an attacker makes some computing or memory resources too busy or too full to handle legitimate requests, or denies legitimate users access to a machine. Examples are Apache2, Back, Land, Mailbomb, SYN Flood, Ping of death, Process table, Smurf.
2. **Remote to user (R2L):** In this type of attack an attacker who does not have an account on a remote machine sends packets to that machine over a network and exploits some vulnerability to gain local access as a user of that machine. Examples are Dictionary, Ftp_write, Guest, Imap, Named, Phf, Sendmail, Xlock.
3. **User to root (U2R):** In this type of attacks an attacker starts out with access to a normal user account on the system and is able to exploit system vulnerabilities to gain root access to the system. Examples are Eject, Loadmodule, Ps, Xterm, Perl, Fdformat.
4. **Probing:** In this type of attacks an attacker scans a network of computers to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use this information to look for exploits. Examples are Ipsweep, Mscan, Saint, Satan, Imap.

The data set has 41 attributes for each connection record plus one class label, which points out whether the data is a normal one or an attack one. The 41 parameters are listed in Table 1.

Table 1 KDD CUP 99 parameter

No.	Feature	No.	Feature
1	duration	2	protocol_type
3	Service	4	flag
5	src_bytes	6	dst_bytes
7	land	8	wrong_fragment
9	Urgent	10	hot
11	num_failed_logins	12	logged_in
13	num_compromised	14	root_shell
15	su_attempted	16	num_root
17	num_file_creations	18	num_shells
19	num_access_files	20	num_outbound_cmds
21	is_hot_login	22	is_guest_login
23	count	24	srv_count
25	serror_rate	26	srv_serror_rate
27	rerror_rate	28	srv_rerror_rate
29	same_srv_rate	30	diff_srv_rate
31	srv_diff_host_rate	32	dst_host_count
33	dst_host_srv_coun	34	dst_host_same_srv_rate
35	dst_host_diff_srv_rate	36	dst_host_same_src_port_ rate
37	dst_host_srv_diff_host_rate	38	dst_host_serror_rate
39	dst_host_srv_serror_rate	40	dst_host_rerror_rate
41	dst_host_srv_rerror_rate		

b) Feature Selection

As shown in Table 1, forty one parameters (attributes) for each pattern(or record) are too many to be used for intrusion detection as some of them may related to others and do not contribute much in the detection. If all are used, the system complexity will be high. A subset of the parameter set can be used to achieve the similar detection result without exhausting the system in computing. Therefore selecting most important and independent parameters to form the subset is a key process in any IDS. There are some studies on the feature selection of KDD data set. Three popular paradigms of the selection are standing out. They are rough set theory (RS), linear genetic programming (LGP), and multivariate adaptive regression splines (MARS). These selection algorithms are summarized in the following.

Rough set theory is an extension of conventional set theory which supports approximations in decision making [21]. It is a mathematical tool for decision support and suits well for the classification of objects. The main contribution of the rough set is to reduce the attribute size. The work of Zainal et al. [21] has shown that only six parameters are useful in the KDD Cup 99 by using the rough set theory. The parameters are: NO. 3, 4, 5, 24, 32, 41 in table 1.

Linear genetic programming (LGP) is a variant of the Genetic Programming (GP) technique that acts on linear genomes [22]. The LGP selection procedure can put the lowest selection pressure on the individuals by allowing only two individuals to participate in a tournament [23]. The work of Sung et al. [22] showed that the six useful parameters of KDD Cup 99 are: NO. 3, 5, 12, 27, 31, 35 in table 1.

Multivariate Adaptive Regression Splines (MARS) is a form of regression analysis introduced by Jerome Friedman in 1991 [24]. It excels at finding optimal variable transformations and interactions, and the complex data structure that often hides in high-dimensional data [25]. The work of Sung et al. [22] showed that the six useful parameters of KDD Cup 99 are: No. 5, 24, 27, 33, 34, and 35 in table 1.

These selection algorithms lead to different feature subset for detection. They are all tested in this project and the results will be discussed in the later section.

c) Parameter Quantization

As shown in Table 1, the KDD Cup 99 features are in one of the following formats, i.e. continuous, discrete, or symbolic. To prepare the parameters in the detection subset for AIS, they should be quantized or normalized. For symbolic features such as protocol_type (3 symbols), service (70 symbols), and flag

d) The definition of immune elements in AIS

Antigens (Ag): numerical character strings with l elements, where l is the number of features selected from the dataset. Ag contains two subsets which are Self (normal patterns) and Nonself (abnormal patterns):

$$Ag = \{Self, Nonself\} \quad (1)$$

$$Self \cap Nonself = \phi \quad (2)$$

Detectors (Antibodies): The Antibodies *Ab* should have the same number of elements as the antigens *Ag. Ab is expected to be representatives of all Nonself.*

Affinity: The measurement to judge the matching between two patterns. Generally, distance is used to measure the affinity of two patterns. The shorter the distance, the closer these two patterns are in a defined l_D space.

In our project, a normalized Mahhatan distance is used for its simplicity. It is defined as following.

$$D(A,B) = \frac{1}{L}\sum_{i=1}^{l}\left|\frac{a(i)-b(i)}{r(i)}\right| \quad (3)$$

Where A = {a(1), a(2) … a(L)} , B={b(1), b(2), … b(L) } are the two patterns to be measured.

R= {r(1), r(2), … r(L)}, where r(i) represents the range of the ith parameter in the detection feature subset.

Two thresholds are defined. Ta is the threshold, used for detector set generation in the negative selection algorithm. Let *X*∈ *Self, Y* is a pattern generated randomly, If $D(X,Y) <$ Ta, then A and B are considered matching and B will be rejected. Otherwise B will be added to the detector set *Ab*. The second threshold, Td, is for live detection. Whenever a live pattern matches any of the patterns in Ab, the alarm will be raised.

IV. RESULT AND DISCUSSION

The raw dataset which we used to generate detectors contains about five million connection records, 700 million bytes. Meanwhile, the testing data we choose contains 300,000 records, and about 45 million bytes.

Three feature selection algorithms have been tested in our experiment. The results are shown in Table 2. TP (true positive) represents that an abnormal pattern is successful detected. FN (false negative) means an abnormal pattern is falsely recognized as a normal pattern. FP (false positive) means that the normal data is mistakenly detected as an abnormal pattern (i.e. an attack) and this is a false alarm. TN (true negative) means that a normal pattern is recognized as a normal one by our system. As shown in Table 2, different feature selection algorithms lead to different detection accuracies. The MARS algorithm has the best attack detect high detection rate in [14] does not perform well in this AIS-based IDS.

Table 2 accuracy of the system

Algorithm	TP	FN	TP rate	FP	TN	TN rate
Rough set	198208	41029	82.85%	7747	53016	87.25%
LGP	189165	50072	79.07%	368	60395	99.394%
MARS	235627	3610	98.491%	3394	57369	94.414%

Some published IDS have achieved TP up to 99.743%, TN up to 89.95% [14] [16]. Compared with these IDS, this AIS-based intrusion detection system has shown that promising results as shown in the table. Fine tuning the algorithm in feature selection and parameter quantization could lead to further improvement on detection rate and complexity.

V. CONCLUSINO AND FUTRUE WORK

In this paper an artificial immune system based intrusion detection system is presented. Negative selection is the algorithm used in the AIS. A number of feature selection algorithms have been tried and compared in experiments using the KDD Cup 99 dataset. The parameter quantization proposed is aiming to reduce the complexity and maintain the detection performance. The system has shown excellent detection accuracy. The experiments shows MARS feature selection algorithm has the best detection accuracy. In the future work, an adaptive mechanism will be introduced to AIS, so that the detector set will be adaptively updated so that the system can adapt to changes in the network situation. Also more dataset will be tried for the performance testing and verification. Other future works include searching and exploring new feature selection algorithms and computing optimization.

REFERENCES

[1] Rebecca Copeland, Converging NGN Wireline and Mobile 3G Network with IMS, Taylor & Francis Group, U.S.A, 2009

[2] Michael T.Hunter, Russell J.Clark, Frank S. Park, "Security Issues with the IP Multimedia Subsystem (IMS): A White Paper",

[3] Hans-Peter Kriegel, Peer Kröger, Arthur Zimek (2009). "Outlier Detection Techniques (Tutorial)". 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2009) (Bangkok, Thailand). http://www.dbs.ifi.lmu.de/Publikationen/Papers/tutorial_slides.pdf. Retrieved 2010-06-05.

[4] Steven A. Hofmeyr and S. Forrest,"Architecture for an Artificial Immune System", Evolutionary Computation Journal, pp. 443-473, 2000.

[5] Leandro N. de Castro and Jonathan Timmis, "Artificial Immune Systems: A New Computational Intelligence Approach", Springer, 2002.

[6] Forrest, S.; Perelson, A.S.; Allen, L.; Cherukuri, R. (1994). "Self-nonself discrimination in a computer" (PDF). Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy. Los Alamitos, CA. pp. 202–212. http://www.cs.unm.edu/~immsec/publications/virus.pdf.

[7] 3GPP Technical Specification of Security: http://www.3gpp.org/ftp/Specs/html-info

[8] 3GPP TS 33.203: Third Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G securtiy; Access security for IP-based services (Release 7).

[9] 3GPP TS 33.210: Third Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G securtiy; Network Domain Security; IP network layer security (Release 7).

[10] Chi-Yuan Chen, Tin-Yu Wu, Yueh-Min Huang, Han-Chieh Chao, "An Efficient end-to-end security mechanism for IP multimedia subsystem", Computer Communications archive. Volume 31, Issue 18, December 2008, page: 68-81.

[11] S. A. Hofmeyr and S. Forrest, "Immunity by design: An artificial immunb system," in Proceedings of the Genetic and Evolutionary Computation Conference. San Mateo, CA: Morgan Kaufmann, July 1999, pp.1289–1296.

[12] D. Dasgupta. Immunity-based intrusion detection systems: A general framework. presented at 22nd Nat. Information Systems Security Conf.. [Online]. Available: http://csrc.nist.gov/nissc/1999/proceedings/papers/p11.pdf

[13] R.P.Lippmann, D.J.Fried, I.Graf, J.W.Haines, K.R.Kendall, D.McClung, D.Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman, "Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation," discex, vol. 02, p. 1012, 2000.

[14] Anazida Zainal, Mohd Aizaini Maarof and Siti Mariyam Shamsuddin, "Feature Selection Using Rough Set in Intrusion Detection", TENCON 2006. 2006 IEEE Region 10 Conference, 14-17 Nov. 2006, pp.1-4

[15] Gursel Serpen and Maheshkumar Sabhnani, "Measuring similarity in feature space of knowledge entailed by two separate rule sets", Knowledge-Based Systems, Volume 19, Issue 1, March 2006, pp. 67-76.

[16] A.H. Sung, and S. Mukkamala, "The Feature Selection and Intrusion Detection Problems". Springer Verlag Lecture Notes Computer Science 3321. 2004, pp. : 468-482.

[17] Cristianini, N., Taylor, S.J, "An Introduction to Support Vector Machines." Cambridge University Press (2000)

[18] Friedman, J.H, "Multivariate Adaptive Regression Splines." Annals of Statistics; Vol.19. (1991) 1-141

[19] Banzhaf, W., Nordin, P., Keller, E.R., Francone, F.D, "Genetic Programming: An Introduction on the Automatic Evolution of Computer Programs and its Applications." Morgan Kaufmann Publishers, Inc (1998)\

[20] Wanli Ma, Dat Tran, and Dharmendra Sharma, ''Negative seleciton with antigen feedback in intrusion detection'', Lecture Notes in Computer Science, 2008, Vol 5132/2008, 200-209.

[21] Anazida Zainal, Mohd Aizaini Maarof and Siti Mariyam Shamsuddin, ''Feature selection using rough set in intrusion detection'', Tencon 2006. 2006 IEEE Region 10 Conference, pp.1-4.]

[22] Banzhaf, W., Nordin, P., Keller, E.R., Francone, F.D. "Genetic Programming: An Introduction on the Automatic Evolution of Computer Programs and its Applications." Morgan Kaufmann Publishers, Inc (1998)

[23] A. H. Sung, and S. Mukkamala, *"The Feature Selection and Intrusion Detection Problems". Springer Verlag Lecture Notes Computer Science 3321*. 2004, pp. : 468-482.

[24] Friedman, J. H. (1991). "Multivariate Adaptive Regression Splines". Annals of Statistics 19 (1): 1–67.

[25] Steinberg, D., Colla, P.L., Kerry. "MARS User Guide." Salford Systems, San Diego (1999)

[26] Hofmeyr, S.. and Forrest S . "Architecture far ao Ariificial Immune System", Evolutionary Computation, 7(1):45-68. (1999).

[27] Gerry Doziert, Douglas Brownf, John Hurley, Krystal Cainf , "Vulnerability analysis of AIS-based intrusion detection systems via genetic and particle swarm red teams" Evolutionary Computation, 2004. CEC2004. 19-23 June 2004, page: 111 - 116 Vol.1

[28] Jungwon Kim and Peter J. Bentley, "Towards an artificial immune system for network intrusion detection: an investigation of clonal selection with a negative selection operator", Evolutionary Computation, 2001. pp, 1244 - 1252 vol. 2, 2001

[29] J.D. Farmer, N. Packard and A. Perelson, (1986) "The immune system, adaptation and machine learning", Physica D, vol. 2, pp. 187–204

[30] H. Bersini, F.J. Varela, Hints for adaptive problem solving gleaned from immune networks. Parallel Problem Solving from Nature, First Workshop PPSW 1, Dortmund, FRG, October, 1990.

Appendix B

An improved Artificial Immune System based Network Intrusion Detection by Using Rough Set

Junyuan Shen
RMIT University
Melbourne, Australia
junyuan.shen@student.rmit.edu.au

Jidong Wang
RMIT University
Melbourne, Australia
jidong.wang@rmit.edu.au

Hao Ai
NUPT University
Nanjing, China
aihao_beibei@163.com

Abstract—**With the increasing worldwide network attacks, intrusion detection (ID) has become a popular research topic in last decade. Several artificial intelligence techniques such as neural networks and fuzzy logic have been applied in ID. The results are varied. The intrusion detection accuracy is the main focus for intrusion detection systems (IDS). Most research activities in the area aiming to improve the ID accuracy. In this paper, an artificial immune system (AIS) based network intrusion detection scheme is proposed. An optimized feature selection using Rough Set (RS) theory is defined. The complexity issue is addressed in the design of the algorithms. The scheme is tested on the widely used KDD CUP 99 dataset. The result shows that the proposed scheme outperforms other schemes in detection accuracy.**

Keywords- Intrusion Detection, Negative selection, Artificial Immune System, KDD CUP 99

I. INTRODUCTION

Driven by the rapid growth of the computer network technologies, the security of the computer and network information is becoming increasingly important. The appearance of the new access technologies and the advanced devices has increased the possibilities of malicious attacks or service abuse by various hackers. Also, with the appearance of multimedia services (video, audio, image, text, etc.), a faster, short-delay anti-virus system is required. However, the traditional passive defence mechanisms like encryptions and firewalls cannot fully meet current security requirements. Therefore, a special attack and misuse detection system is needed. The intrusion detection system (IDS) is such a system, which is composed by a series of devices and software applications to monitor network activities in order to protect the system from malicious activities.

The general IDS detect unauthorized users or processes by comparing a user's behaviour with the user's profile. Two approaches, misuse detection and anomaly detection, are usually used in the intrusion detection process. The misuse detection is used to detect the intrusion when the behavior of the system matches with any of the intrusion signatures in the user profile. And the anomaly detection, which is also called as outlier detection [1], is used to detect the intrusion when the given data set does not match with the established normal behavior.

Various techniques have been used for building IDS, like Support Vector Machines (SVM) [2], Multivariate Adaptive Regression Splines (MARS) [3], and Linear Genetic Programming (LGP) [4], etc. Some of them give good performance in specific attack areas, while they might not detect other attacks well. In recent years, bio-inspired algorithms have been studied and applied in intrusion detection [5] aiming for better performance. Algorithms such as Genetic Algorithm (GA), Artificial Neural Networks (ANN) and Artificial Immune Systems are widely studied. AIS is a relatively new comer among them. The concept of AIS was proposed in mid 1980s. Farmer, Packard and Perelson [6], Bersini and Varela's [7] work have started the area. AIS has not become a subject of its own untill mid 90s. It has been defined as: "Adaptive systems inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving." by Castro and Timmis [8]. Early works of AIS based IDS can be found in [9]. A Multilayer IDS using AIS was proposed by Dasgupta [10] in order to provide systematic defense. These AIS based IDS have achieved good detection results. However, their computing complexity is quite high due to the complicated feature comparing. While, for IDS, responding time is also an important issue. The more complexity the system, the more computing time and the longer responding time will be. Large parameter set in IDS can increase the detection accuracy. However, the more parameters using, the more complex the system will be. So, the trade off between the complexity and the accuracy is a challenge. Our study on AIS based IDS is to further improve its detection accuracy while keeping a low algorithm complexity.

In this paper, an improved AIS based intrusion detection system with Rough Set feature selection algorithm is presented. The anomaly detection in the system is set up based on AIS negative selection algorithm. And the feature selection algorithm is used to reduce the complexity of the system. The artificial immune system and the negative selection algorithm are introduced in Section II. The AIS based IDS is presented in Section III. Our experiment and results are illustrated in Section IV. Section V draws a conclusion and some future works are discussed.

II. ARTIFICIAL IMMUNE SYSTEM

Artificial Immune System (AIS) applies to various areas of researches that attempt to build a bridge between immunology and engineering by using the techniques of mathematical and computational modeling of immunology.

The origin of AIS is rooted in the early theoretical work of J.D. Farmer, N.H. Packard, A.S. Perelson [11,12],F. Varela, A. Coutinho, B. Dupire, and N. Vaz [13]. It was first proposed in mid 1980s and became a subject of its own in mid 90s. Originally, AIS aimed to find efficient abstractions of processes in the immune system [14]. By carefully reviewing the efficient natural mechanism, a number of computer scientists proposed artificial immune based computer models to solve various problems ranging from virus detection, fault analyzing to clustering. Two researchers, Hugues Bersini and Stephanie Forrest, played an important role in crossing the divide between computing and immunology. Bersini and Forrest did a lot of basic works rooted from immunology and their works formed a solid foundation of the area of AIS. With regards to Bersini, he was focusing on the basic theory of immune network and examining how the immune system maintained its memory and how to build a model to mimic that progress. And for Forrest, she was focusing on the application area of the AIS. She proposed the idea of introducing the immune system into the computer security area by using its ability to distinguish between self and non-self.

Negative selection, which is proposed by Forrest et al. [15], is inspired from the negative selection process of the adaptive immune system [16]. The important characteristic of the human immune system is that it can maintain its diversity and generality, and it can detect a large number of antigens by using a small number of antibodies. In order to make it possible, several functions will be processed [17]. One of those functions is to develop the antibodies through the gene library. The gene library will be used in creating thymus cell (T cell) and bone marrow cell (B cell). While creating a new antibody, the gene segments in the gene library will be randomly selected and assembled. As shown in Figure 1, large number of antibodies can be generated from combining different genes segments in the gene library.

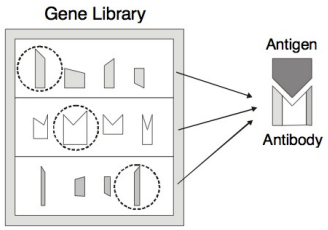

Figure 1. Gene Expression Process [27]

However, there is a problem due to the full immune response above. Not only responding to harmful antigens, those new generated antibody may also react to self-cells coming from the host. In order to protect the body from self-reactive, the human immune system produces the negative selection.

In the case of an anomaly detection domain, the algorithm prepares a set of exemplar pattern detectors trained on normal (non-anomalous) patterns that model and detect unseen or anomalous patterns [18]. The principle of the negative selection is shown in Figure 2.

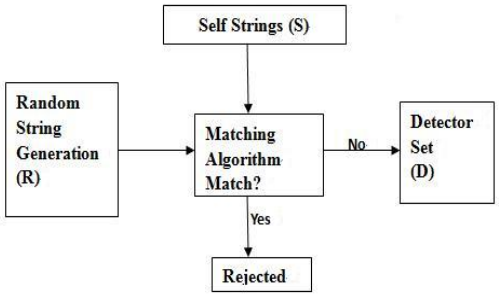

Figure 2. The Principle of Negative Selection

As shown in Figure 2, the basic idea of the negative selection is to generate a selected detector set D and use the detector set to distinguish the new data. In the process, a set of detectors R will be randomly generated, and all the randomly generated detectors will be compared with each elements of the self-string set S. Under certain matching algorithms, if the detector in set R fails to match any element in S, it will be saved in the detector set D, otherwise, it will be rejected.

In the matching process, several algorithms have been proposed to determine the difference between self and non-self like Eucliden distance, hamming distance, r-contiguous bit rule algorithm, etc. In this paper, the Manhattan Distance will be used because of its simplicity. The affinity (difference) of the set R and S are related to the distance between them. The definition of the distance is shown as follows

Let $R=<R_1, R_2 \ldots R_m>$, $S=<S_1, S_2 \ldots S_m>$,

$$\text{Manhattan } D = \sum_{i=1}^{m} |R_i - S_i|$$

In the intrusion detection process, for any pattern to be checked, it needs to be compared with all the patterns in the detector set. If it matches to any pattern in the detector set it will be considered as a non-self element, otherwise, it will be considered as self. AIS has been found applications in many areas such as optimization, data analysis, machine learning, pattern recognition, etc and network intrusion detection which is the focus of this paper.

III. AIS BASED IDS

Generally, network intrusion detection is based on the examination of monitored network parameters. Different examine algorithms lead to different IDS. The general AIS based IDS [20] [21] [22] can be divided into two parts, i.e. detector set generation and the live detection. To form the detection set, negative selection algorithm is applied, as discussed in Section II. In the live detection stage, a monitored network parameter pattern is compared with

A compact and effective detector set can reduce the algorithm computing complexity. For detectors that do not contribute any detection in a period of time, they should be removed or put to a sleeping state. Therefore, all the mature detectors will have a time_to_live (TTL) parameter. Whenever detection is occurred, all detectors' TTLs will be deducted by one except for the detector which detects the intrusion. Its TTLwill be reset to the maximum. When a detector reaches its lifetime, ie its TTL becomes zero; this detector will be become inactive.

a) The definition of immune elements in AIS

Antigens (Ag): numerical character strings with L elements, where L is the number of features selected from the dataset. Ag contains two subsets that are Self (normal patterns) and Nonself (abnormal patterns):

$$Ag = \{Self, Nonself\} \quad (1)$$

$$Self \cap Nonself = \phi \quad (2)$$

Detectors (Antibodies): The Antibodies *Ab* should have the same number of elements as the antigens *Ag*. *Ab is expected to be representatives of all Nonself.*

Affinity: The measurement to judge the matching between two patterns. Generally, distance is used to measure the affinity of two patterns. The shorter the distance, the closer these two patterns are in the defined L space.

In our project, a normalized Mahhatan distance is used for its simplicity. It is defined as following.

$$D(A,B) = \frac{1}{L}\sum_{i=1}\left|\frac{a(i)-b(i)}{r(i)}\right| \quad (3)$$

Where A = {a(1), a(2) … a(L)} , B={b(1), b(2), … b(L) } are the two patterns to be measured.

R= {r(1), r(2), … r(L)}, where r(i) represents the range of the ith parameter in the detection feature subset.

Two thresholds are defined. Ta is the threshold, used for detector set generation in the negative selection algorithm. Let $X \in Self$, *Y* is a pattern generated randomly, If $D(X, Y) <$ Ta, then A and B are considered matching and B will be rejected. Otherwise B will be added to the detector set *Ab*. The second threshold, Td, is for live detection. Whenever a live pattern matches any of the patterns in Ab, the alarm will be raised. In our scheme, different Td has been tested to find a trade-off between the attack detection accuracy and false alarm rate.

b) Parameter Quantization

As shown in Table 1, the KDD Cup 99 features are in one of the following formats, i.e. continuous, discrete, or symbolic. To prepare the parameters in the detection subset for AIS, they should be quantized or normalized. For symbolic features such as protocol_type (3 symbols), service (70 symbols), and flag

IV. KDD CUP 99 WITH ROUGH SET THEORY

The data set used in our experiment is the KDD Cup 99 data set, which is the most widely used data set for network-based intrusion detection. This data set is built based on the data captured in DARPA'98 IDS evaluation program [23]. The data set contains 24 training attack types and 14 additional attack types in the test data only. It has 41 parameters in each data record and the data type is shown as follows:

- 0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,8,8, 0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00,0.11,0.00 ,0.00,0.00,0.00,0.00,normal.
- 0,icmp,ecr_i,SF,1032,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,511, 511,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,255,1.00,0.00, 1.00,0.00,0.00,0.00,0.00,0.00,smurf.

Each parameter in the data string has its own meaning which is shown in Table 1. The complexity is so high if all the 41 parameters are used and the responding time of the IDS will be slow. A feature selection is needed to minimize the data set.

Table 1 KDD CUP 99 parameter

No.	Feature	No.	Feature
1	duration	2	protocol_type
3	Service	4	flag
5	src_bytes	6	dst_bytes
7	land	8	wrong_fragment
9	Urgent	10	hot
11	num_failed_logins	12	logged_in
13	num_compromised	14	root_shell
15	su_attempted	16	num_root
17	num_file_creations	18	num_shells
19	num_access_files	20	num_outbound_cmds
21	is_hot_login	22	is_guest_login
23	count	24	srv_count
25	serror_rate	26	srv_serror_rate
27	rerror_rate	28	srv_rerror_rate
29	same_srv_rate	30	diff_srv_rate
31	srv_diff_host_rate	32	dst_host_count
33	dst_host_srv_coun	34	dst_host_same_srv_rate
35	dst_host_diff_srv_rate	36	dst_host_same_src_port_ rate
37	dst_host_srv_diff_host_rate	38	dst_host_serror_rate
39	dst_host_srv_serror_rate	40	dst_host_rerror_rate
41	dst_host_srv_rerror_rate		

Rough Set Theory (RST), first proposed by Polish computer scientist Zdzisław I. Pawlak, is an extension of conventional set theory that supports approximations in decision-making [19]. It is a mathematical tool for decision support and suits well for the classification of objects. A lot of researches have been focused in the RST-based machine

tasks.

The work of Zhang et al. [25] has shown that RST showed high detection accuracy and feature ranking was fast in determining the categories of the attacks in IDS. And Zainal et al. [26] has shown that the IDS have performed well by using RST and the six highest rank features by RST were Service, flag, src_bytes, srv_count, dst_host_count, dst_host_srv_rerror_rate in table 1. But unfortunately, the false alarm rate in Zainal's research is relatively high.

In our scheme, an improved rough set theory is introduced. By using the six parameters chosen from the KDD Cup 99 data set, each parameter is associated with a 'weight'. The weights for the six parameters are different because of the different contributions of these parameters to the system performance. A range of the weights have been tested in our experiment in order to find a suitable one for the AIS based intrusion detection system.

V. RESULT AND DISCUSSION

The raw dataset that we used to generate detectors contains about five million connection records, 700 million bytes. Meanwhile, the testing data we choose contains 300,000 records, and about 45 million bytes. In our scheme, as described in Section IV, different parameters chosen by the Rough set need to give different weight. Before a weight is finalized, an "influence factor" is tested for each parameter.

Originally, an AIS based intrusion detection system is built based on the C++ platform. In the negative selection process, each parameter has a weight of '6' for all the six parameters chosen based on rough set theory, and the total weight is equal to 36. Then, one parameter will change by the step size of 1 and the other five will change by the step size 0.2, which keep the total weight of the equation 36 unchanged. According to the test data we use in the KDD Cup 99 data set, 239237 attacks are contained. And the attack detection quantity for different parameters is shown in the Figure 3. The

As shown in Figure 3, by changing the weight of each single parameter and keep the others the same, the attack detection number will change in the meantime. The table 2 below shows the different attack detection accuracy for each single changed parameter.

Table 2 Attack detection range for each single changed parameter

Parameter Type	service	flag	Src_bytes	Src_count	dst_host_ count	Dst_host_srv_ rerror_rate
Attack Detection Range	185843 to 239237	187200 to 237280	169610 to 239237	186953 to 239237	181493 to 239237	186076 to 239237

order to find a best parameter weight combination for the rough set theory, the exhaustive method is used. All the combination of the chosen parameters (service, src_bytes and dst_host_count) is tested and the detection accuracy is shown in Figure 4.

In Figure 4, Series 1 represents the attack detection accuracy, and Series 2 represents the normal detection accuracy. As shown in Figure 4, the system detection accuracy shows a significant improvement with different weighting factors of the parameter. The true positive rate (TP rate) can up to 98.25% (with TN rate 99.90%), and the true negative rate (TN rate) up to 99.97 (with TP rate 82.03%).

In general, compared with the original rough set based IDS [26], by introducing the "weight" scheme, the proposed IDS provided a better TN rate (above 99% compared with 89.95%), and relatively high TP rate. Fine tuning the algorithm in feature selection and parameter quantization could lead to further improvement on detection rate and complexity.

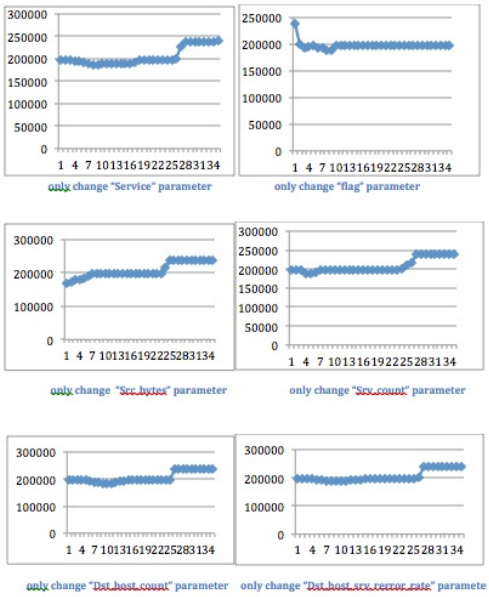

Figure 3. Attack Detection Quantity

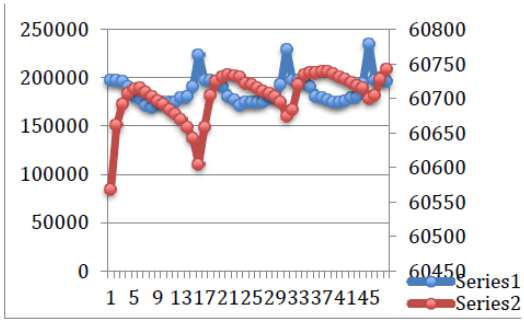

Figure 4. System Detection Accuracy

VI. CONCLUSINO AND FUTRUE WORK

In this paper an improved artificial immune system based intrusion detection system by using rough set is presented. In order to find a best combination of the six parameters chosen by rough set theory, a number of tests have been conducted. The results are compared using the KDD Cup 99 dataset. The rough set theory proposed is aiming to reduce the complexity and maintain the detection performance. The system has shown excellent detection accuracy. The improved rough set theory can significantly increase the TN rate, and keep relatively high TP rate in the meantime. For future work, an adaptive mechanism will be introduced to AIS, so that the detector set will be adaptively updated so that the system can adapt to changes in the network situation. Also more feature selection algorithms can be tried for the performance improvement and verification.

REFERENCES

[1] Hans-Peter Kriegel, Peer Kröger, Arthur Zimek (2009). "Outlier Detection Techniques (Tutorial)". 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2009) (Bangkok, Thailand).

[2] Cristianini, N., Taylor, S.J, "An Introduction to Support Vector Machines." Cambridge University Press (2000)

[3] Friedman, J.H, "Multivariate Adaptive Regression Splines." Annals of Statistics; Vol.19. (1991) 1-141

[4] Banzhaf, W., Nordin, P., Keller, E.R., Francone, F.D, "Genetic Programming: An Introduction on the Automatic Evolution of Computer Programs and its Applications." Morgan Kaufmann Publishers, Inc (1998)

[5] S. A. Hofmeyr and S. Forrest, "Immunity by design: An artificial immunb system," in Proceedings of the Genetic and Evolutionary Computation Conference. San Mateo, CA: Morgan Kaufmann, July 1999, pp.1289–1296.

[6] J.D. Farmer, N. Packard and A. Perelson, (1986) "The immune system, adaptation and machine learning", Physica D, vol. 2, pp. 187–204

[7] H. Bersini, F.J. Varela, Hints for adaptive problem solving gleaned from immune networks. Parallel Problem Solving from Nature, First Workshop PPSW 1, Dortmund, FRG, October, 1990.

[8] Leandro N. de Castro and Jonathan Timmis, "Artificial Immune Systems: A New Computational Intelligence Approach", Springer, 2002.

[9] S. A. Hofmeyr and S. Forrest, "Immunity by design: An artificial immunb system," in Proceedings of the Genetic and Evolutionary Computation Conference. San Mateo, CA: Morgan Kaufmann, July 1999, pp.1289–1296.

[10] D. Dasgupta. Immunity-based intrusion detection systems: A general framework. presented at 22nd Nat. Information Systems Security Conf.. [Online]. Available: http://csrc.nist.gov/nissc/1999/proceedings/papers/p11.pdf

[11] J.D. Farmer, N.H. Packard, A.S. Perelson, The immune system, adaptation, and machine learning, Physica D 22 (1986) 187–204.

[12] A.S. Perelson, Immune network theory, Immunological Reviews 110 (1989) 5–36.

[13] F. Varela, A. Coutinho, B. Dupire, N. Vaz, Cognitive networks: Immune, neural and otherwise, Theoretical Immunology 2 (1988) 359–375.

[14] Tizard, I. R., 1995, Immunology: Introduction, 4th Ed, Saunders College Publishing.

[15] Steven A. Hofmeyr and S. Forrest,"Architecture for an Artificial Immune System", Evolutionary Computation Journal, pp. 443-473, 2000.

[16] Leandro N. de Castro and Jonathan Timmis, "Artificial Immune Systems: A New Computational Intelligence Approach", Springer, 2002.

[17] Kim, J. and Bentley, P., 1999a, The Human Immune System and Network Intrusion Detection, 7th European Conference on Intelligent Techniques and Soft Computing (EUFIT '99), Aachen, Germany (to appear).

[18] Forrest, S.; Perelson, A.S.; Allen, L.; Cherukuri, R. (1994). "Self-nonself discrimination in a computer" (PDF). Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy. Los Alamitos, CA. pp. 202–212. http://www.cs.unm.edu/~immsec/publications/virus.pdf.

[19] Anazida Zainal, Mohd Aizaini Maarof and Siti Mariyam Shamsuddin, "Feature selection using rough set in intrusion detection", Tencon 2006. 2006 IEEE Region 10 Conference, pp.1-4.]

[20] Hofmeyr, S.. and Forrest S . "Architecture far ao Ariificial Immune System", Evolutionary Computation, 7(1):45-68. (1999).

[21] Gerry Doziert, Douglas Brownf, John Hurley, Krystal Cainf , "Vulnerability analysis of AIS-based intrusion detection systems via genetic and particle swarm red teams" Evolutionary Computation, 2004. CEC2004. 19-23 June 2004, page: 111 - 116 Vol.1

[22] Jungwon Kim and Peter J. Bentley, "Towards an artificial immune system for network intrusion detection: an investigation of clonal selection with a negative selection operator", Evolutionary Computation, 2001. pp, 1244 - 1252 vol. 2, 2001

[23] R.P.Lippmann, D.J.Fried, I.Graf, J.W.Haines, K.R.Kendall, D.McClung, D.Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman, "Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation," discex, vol. 02, p. 1012, 2000.

[24] Banzhaf, W., Nordin, P., Keller, E.R., Francone, F.D, "Genetic Programming: An Introduction on the Automatic Evolution of Computer Programs and its Applications." Morgan Kaufmann Publishers, Inc (1998)

[25] L. Zhang, G. Zhang, L. Yu, J. Zhang, and Y. Bai, "Intrusion Detection Using Rough Set Classification." Journal of Zheijiang University Science. 2004 5(9), pp. 1076-1086.

[26] Anazida Zainal, Mohd Aizaini Maarof and Siti Mariyam Shamsuddin, "Feature selection using rough set in intrusion detection", Tencon 2006. 2006 IEEE Region 10 Conference, pp.1-4.

[27] Kim, J and Bentley, PJ (1999) Negative selection and niching by an artificial immune system for network intrusion detection. In: (Proceedings) Proc. Late-Breaking Papers at the Genetic and Evolutionary Computation Conference (GECCO'99), Orlando, Florida. (pp. 149 - 158).