Copyright ⓒ 2010 KSII

GEP-based Framework for Immune-Inspired Intrusion Detection

Wan Tang[1*], Limei Peng[2], Ximin Yang[1], Xia Xie[3] and Yang Cao[4]
[1] College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China
[e-mail: {tangwan, yangximin}@scuec.edu.cn]
[2] Department of Electrical Engineering, GRID Middleware Research Center, KAIST, Taejon, 305, Korea
[e-mail: aurora_plm@kaist.ac.kr]
[3] College of Computer Science, Huazhong University of Science & Technology, Wuhan 430073, China
[e-mail: shelicy@hust.edu.cn]
[4] School of Electronic Information,Wuhan University, Wuhan 430070, China
[e-mail: caoy_whu@163.com]
*Corresponding author: Wan Tang

Received August 3, 2010; revised October 7, 2010; accepted October 27, 2010; published December 23, 2010

Abstract

Immune-inspired intrusion detection is a promising technology for network security, and well known for its diversity, adaptation, self-tolerance, etc. However, scalability and coverage are two major drawbacks of the immune-inspired intrusion detection systems (IIDSes). In this paper, we propose an IIDS framework, named GEP-IIDS, with improved basic system elements to address these two problems. First, an additional bio-inspired technique, gene expression programming (GEP), is introduced in detector (corresponding to detection rules) representation. In addition, inspired by the avidity model of immunology, new avidity/affinity functions taking the priority of attributes into account are given. Based on the above two improved elements, we also propose a novel immune algorithm that is capable of integrating two bio-inspired mechanisms (i.e., negative selection and positive selection) by using a balance factor. Finally, a pruning algorithm is given to reduce redundant detectors that consume footprint and detection time but do not contribute to improving performance. Our experimental results show the feasibility and effectiveness of our solution to handle the scalability and coverage problems of IIDS.

Keywords: Network intrusion detection, artificial immune system, gene expression programming

A preliminary version of this paper appeared in IEEE IWQoS, June 16-18, 2010, Beijing, China. This version proposes a GDP algorithm and includes a concrete analysis and supporting implementation results on GEP-IIDS. This work is supported by the National Natural Science Foundation of China (60603008) and the Special fund for Basic Scientific Research of Central Colleges (ZZZ10003), South-Central University for Nationalities, China.

DOI: 10.3837/tiis.2010.12.017

1. Introduction

The Internet is becoming an universal communication network for all kinds of information. It is envisaged to support not only current services, but also new services with various traffic characteristics and quality of service (QoS) performance requirements [1][2]. Since network resources, such as buffer and link capacity, are finite and shared by traffic flows, QoS is very sensitive to attacks, especially denial of service (DoS) attacks [3]. A DoS attack aims to deny access to shared services or resources by legitimate users. It can deplete the server system resources such as CPU and memory, exhaust the network resources, and disrupt the traffic transmission.

Intrusion detection is an active defense technology for network security [4]. Although a large body of research has been devoted to intrusion detection, new directions based on computational intelligence approaches, e.g., artificial immune systems(AIS), are being pursued to cope with dynamic and increasingly complex networks more effectively [5][6]. AIS is inspired by the biological immune system (BIS) which has several useful features for intrusion detection system (IDS), such as detection, diversity, adaptation, self-tolerance, etc.

Most of the existing immune-inspired IDSes (or IIDSes) are based on negative selection algorithms in AIS [7][8][9][10][11]. However, such IIDSes have two major drawbacks, namely scalability and coverage [5][12][13]. The scalability problem refers to the fact that the system has to produce a large number of detectors (corresponding to rules for intrusion/attacks detection) to achieve desirable detection rate, and accordingly, it requires huge amounts of temporal and spatial cost. The coverage problem refers to the fact that there still exist holes (malicious attacks which cannot be detected by the system) that are not covered by a complete detector repertoire. These two problems have made the IIDS ineffective [13][14][15].

Recent research on intrusion detection has also investigated the use of gene expression programming (GEP), which is a new computational intelligence technique that takes advantages of the combination of genetic algorithm (GA) and genetic programming (GP) [16]. Applying a modern metaheuristic GEP, a novel approach to detecting web application attacks was proposed by J. Skaruz [17]. A GEP-based constraint grammar and a constraint-based GEP rule extraction algorithm (CGREA) were proposed in our previous work [18]. Within a few evolution generations, CGREA can generate a small number of bi-attribute detection rules (i.e., each rule consists of only two attributes) for IDS to achieve a high degree of coverage. However, in CGREA, the diversity of the generated rule population is still insufficient, and the coverage of many rules overlaps with each other, resulting in low efficiency.

In general, the coverage problem of IIDS can be solved to some extent by increasing the number of detectors. However, the redundant detectors, which do not help significantly in increasing the detection ability, consume more memory and detection time, and further reduce the system scalability. In negative selection algorithms, the greedy and variable-length detector generation methods optimize the detector set during the generation process [19][20]. However, these methods represent the detectors in terms of binary string, *r*-continuous and *r*-trunk bit matching schemes and cannot be applied to the GEP-based rules/detectors.

To improve the existing IIDSes and make them more efficient, in this paper, we propose a new framework named GEP-IIDS. The framwork consists of three modified basic system elements: 1)a representation of artifical immune entities, i.e. antigens and detectors, which are represented respectively based on attribute-gene and constraint-based GEP, 2) avidity/affinity functions considering attribute-priority, and 3) an avidity-model based clonal selection

(AMCS) algorithm inspired by the clonal selection and avidity model of BIS. Additionally, a GEP-based detectors pruning (GDP) algorithm is also proposed to eliminate the redundant detectors for GEP-IIDS.

The rest of the paper is organized as follows. Section 2 briefly describes the analogy between BIS and IDS, introduces the basic elements of IIDS, and discusses the concepts of affinity and avidity. Related works are summarized in Section 3. In Section 4, a new framework, in which the AMCS algorithm is a critical component, is proposed to improve detection performance and reduce complexity for IIDS. Furthermore, the GDP algorithm is also presented. The experimental results are shown in Section 5. Finally, Section 6 concludes the work.

2. Immune-Inspired Intrusion Detection

BIS, which consists of immune organs, immune cells and immune molecules, is a very complex and precise defense system with the intention to protect the body from harmful substances. We will briefly describe the analogy between BIS and IDS using basic terms as follows.

In BIS, antigens can be either self or non-self. They correspond to network access patterns that are either normal or abnormal in IDS. A BIS has many detectors[1] to recognize non-self antigens (for appropriate actions such as killing those antigens), and these detectors correspond to the set of rules used in IDS to detect malicious access patterns (i.e., attacks or intrusion[2]).

Detectors are generated via positive selection and negative selection in BIS. In negative selection (NS), an immature detector will be eliminated if it binds to any self antigen, otherwise it becomes mature and is distributed for detecting non-self antigens. Then BIS can detect non-self antigens without mistakenly detecting any self one. In positive selection, only those detectors that bind to non-self antigens will survive. This corresponds to the toleration of self programs, and the detection of misuse, abuse and unauthorized use of computer networks in IDS.

In BIS, the efficiency of detection is maintained by the evolution of detectors via clonal selection, which always can bind to the dynamically changing antigens. Activating detectors are divided into a number of clones that have the same properties as their parent detectors or mutated properties. The detectors that bind to more non-self antigens have more chance of being selected for cloning, and the new self-reactive clones will be eliminated. The clonal selection corresponds to the learning process adapting to increasing new malicious access patterns in IDS.

IIDSes are based on AIS, which are adaptive computational systems inspired by the principles and processes of BIS. An IIDS framework can be developed containing three basic system elements: representation of artificial immune entities (i.e. antigens and detectors), affinity function and immune algorithm [6][13]. The structure of an immune entity, namely genotype, is arrayed by a series of genes. An appropriate affinity function is the metric to quantify the interactions between two artificial immune entities and determine corresponding matching patterns or rules. Immune algorithm is abstracted from immunological principles and used to generate a set of suitable detectors. Most existing works on IIDS are devoted to the development of immune algorithms inspired by NS [5][6].

[1] We use the term “detector” loosely to refer to some immunologic terms, e.g., T-cell, B-cell, recognition receptor, and antibody.

[2] The terms, “intrusion” and “attack,” are used loosely in this paper.

In the biologic avidity model, also known as quantitative model, the cumulative interactions between antigens and detectors are governed in part by avidity [21]. The degree of interactions leads to positive selection or negative selection by which the fate of each detector is determined. According to the avidity model, positive selection of detectors results from their 'weak' interactions, while their 'stronger' interactions lead to NS for detectors. Moreover, the concepts of affinity and avidity are different in the immunology:

- ***affinity***, constantly describes the strength of interaction between a detector and a single binding site of antigen.
- ***avidity***, is defined as the overall binding strength of a detector to an antigen.

Accordingly, the concepts of affinity and avidity in AIS should be different. However, most of the literatures have not differentiated the concepts between them, and only used affinity to denote the combination intensity between detector and antigen. Therefore, this paper defines ***affinity*** as the similarity of the detected distance between two entities, while ***avidity*** is the objective function for the candidate solutions, or the metric for evaluating the adaptation of candidate solutions.

3. Related Work

Existing research on the application of AIS to intrusion detection is based on three distinct philosophies: immune system conventional algorithms, danger theory and NS. Most of the research on IIDSes focuses on the development of AIS algorithms inspired by NS [5]. In this section, we briefly review the research on NS algorithms and GEP.

The first NS algorithm (NSA) and an IIDS architecture based on AIS called LYSIS were proposed by S. Forrest *et al*. [7]. NSA consists of three phases: definite self-patterns, generate detector patterns and detect abnormality. In the first phase, the profiled normal patterns is regarded as "self". NS is used in the second phase, when a pattern is selected to be a detector if and only if it cannot match any self pattern based on an affinity function. Detectors are represented in the form of binary strings and the affinity between immune entities is quantified using *r*-contiguous bit matching scheme.

Dasgupta and Zhou compared negative characterization to positive characterization, and proposed an NS algorithm called *v*-detectors [8][9]. In *v*-detectors, detectors modeled through hypersphere with various radii are used in an efficient manner to achieve maximum coverage. The affinity function of *v*-detectors is defined based on the Euclidean distance.

Kim and Bentley introduced a dynamic clonal selection algorithm (DynamiCS) which combined NS and clonal selection for IIDS [10][11]. DynamiCS evolves detectors which classify non-self from self. Simple binary genotype representation is used to encode the conjunctive-rule-based detectors consisting of a number of genes each representing an attribute of detector phenotype. An antigen matches with a detector if all their existing genes match. The affinity between a single detector and a non-self sample (i.e. antigen) is based on the match count.

Although NS is the most popular immune algorithm used in IIDS, there are two drawbacks: scalability and coverage, as mentioned earlier. In fact, T. Stibor *et al*. stated that NS is *not* appropriate for network IDS [12][13][14][15]. Nevertheless, Dasgupta *et al*. believed that there are many aspects of the NS algorithm that worth further exploration [6][9][22].

Fortunately, the algorithms inspired by clonal selection are more scalable than the NS algorithms, in which a small set of best individuals is maintained so as to solve the problem using minimal resources. Based on the clonal selection and affinity maturation theories, the

CLONALG algorithm is proposed [23], and the structure based on the immune clonal selection algorithm (ICSA) is designed for intrusion detection [24].

Besides the affinity functions of most immune algorithms, the exsiting GEP fitness functions, are not adaptable to be the avidity functions for IIDS. More specifically, calculating fitness according to matching ratio is the most frequently used approach. However, it is too simple to reflect the recognition status of normal data (i.e. self antigens). The functions, considering the total recognition capacity in terms of specificity and sensitivity, are only adaptive to the classification recognition.

Although the best rules generated via the CGREA algorithm are more effective than those generated by some approachs based on other computational intelligence approaches (e.g., Decision Trees and Neural Networks) [17], however, we note that there are still some disadvantages. In CGREA, the diversity of the GEP-rule population is still insufficient, and the degree of coverage-overlap among rules is high. Furthermore, the fitness function, in which the completeness (coverage probability of attacks) and consistency (the approximate degree of distributions of training set and covered set) of the detector are incorporated, cannot reflect the match degree between a detector and a single training record.

In most of the immune algorithms, including AMCS and CGREA, the number of detectors is set to a fixed value based on preliminary experience, and this will bring about redundant detectors. These redundant detectors consume footprint and detection time, but do not contribute to improving detection performance. To the best of our knowledge, no other IIDS has a separate module to handle the redundant detectors. Instead, the redundant detectors are recognized and deleted during generating process. For example, the greedy generation scheme is based on the *r*-continuous bit matching: when generating an initial detector, it searches for any detector that can match the initial one. If there exists the detector, it means that the initial detector is redundant and should be deleted [18]. Meanwhile, the variable-length method that generating variable-length detectors can solve the problem with “holes” caused by the *r*-continuous bit matching [20], and the detector set is optimized during the detector generation process to avoid the redundant detectors.

In the above mentioned methods, judging whether a detector is redundant or not is done during its generation time. That is, when an initial detector is generated, it is required to match with all existing detectors. The time complexity is proportional to the product of the number of generated initial detectors and the required matching times for each detector. Accordingly, the overhead increases with the number of detectors. Moreover, these optimizing strategies with a high time complexity were proposed for the detectors represented in terms of binary strings. Therefore, they are not adaptive to optimize the GEP-based detector set.

4. GEP-based IIDS Framework

To address the above mentioned issues in existing IIDSes, a GEP-based framework is proposed in this section. We will first describe how to represent the antigens and detectors. We then define new avidity/affinity functions considering attribute-priority, describe the proposed avidity-model based on clonal selection (AMCS) algorithm, and finally propose a detector-pruning algorithm for the framework. Our work differs from the existing clonal selection algorithms and NS algorithms in considering both NS and positive selection. Need to mention that although our work uses CGREA to represent the detectors, it also has several novel aspects and critical improvement when compared to our previous works on GEP.

4.1 Representation of Artificial Immune Entities

(1). Antigen Representation

The proposed antigen genotype is shown in **Fig. 1**. The attribute set of antigen represents the abstraction of network behaviors, and each antigen is composed of *n* attribute genes and classified as ***non-self*** and ***self***. The non-self antigens and self antigens correspond respectively malicous network behaviors and normal behaviors.

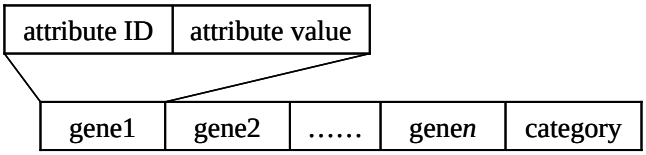


Fig. 1. Antigen genotype

***Definition* 1:** An antigen is denoted as follows:

$$I_{antigen} = \{(v_1, g_1), (v_2, g_2), \ldots, (v_n, g_n), category\}, \quad (1)$$

where (v_i, g_i) denotes the i-th (i =1, …, n) attribute gene, v_i and g_i are the ID and the value of the attribute gene respectively, and category$\in$ {*self*, *non-self*}.

The time-continuous attack packets are interrelated, and the separate attributes of packets are incapable of describing the network behaviors well. In this paper, the attributes of antigen correspond to the derived features of KDD CUP'99 DATA set (here in after to be referred as KDD'99 set), which can represent the interrelated network using the time-based traffic features [25].

Any network connect record in the KDD'99 set contains 41 features, and is labeled as either normal, or an attack. If a record is normal, the category of its corresponding antigen is *self*, or else is *not-self*. Acorrdingly, we set attribute ID *ai* (*i*= 1,…, 41) to correspond to the *i*-th feature of KDD'99 records, e.g., *a*1, *a*2 and *a*3 correspond to feature: "duration," "protocol-type," "service," respectively. Two records of KDD'99 set and corresponding antigens are given below as examples:

- Example1

KDD'99 connection record:
0,tcp,http,SF,241,259,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0.00,
0.00,0.00,0.00,1.00,0.00,0.00,14,149,1.00,0.00,0.07,0.04,0.00, 0.00,0.00,0.00,normal.

antigen:
{(a1,0), (a2,tcp), (a3,http), (a4,SF), (a5,241), (a6,259), …, (a23, 1), …, (a39, 0.00), (a40,0.00), (a41,0.00), ***self***}

- Example2

KDD'99 connection record:
0,icmp,ecr_i,SF,132,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,511,511,
0.00,0.00,0.00,0.00,1.00,0.00,0.00,168,23,0.14,0.02,0.14,0.00, 0.00,0.00,0.00,0.00,smurf.

antigen:
{(a1,0), (a2,icmp),(a3,ecr_i),(a4,SF),(a5,132),(a6,0), … , (a23, 511), …, (a39, 0.00), (a40,0.00),(a41,0.00), ***non-self***}

(2). Detector representation

In this paper, a detector is represented differently from an antigen, and mainly includes the following fields:

1) Genotype ($\boldsymbol{I_{detector}}$), the encoding of a detector, which is composed of several genes. Each of these genes of a detector corresponds to one attribute of network behaviors. This genotype is capable of improving the detector's coverage by using relationship symbols, such as " >," "!=" and "≤".

2) Self avidity ($\boldsymbol{avidity_{self}}$), the value of binding strength with self antigens;

3) Non-self avidity ($\boldsymbol{avidity_{non\text{-}self}}$), the value of binding strength with non-self antigens.

As being briefly mentioned earlier, in IIDS, a detector is the rule for recognizing attacks. The constraint-based GEP rule (here in after to be referred as GEP-rule), has been proposed in our previous work and been proved to be feasible and effective [18]. Let L be a logical operator set, R be a relational operator set, and A be an arithmetic operator set. Note that L, R and A all are subsets of the function set F_s. The terminal set includes attribute set A_s and constant set C_s. Then, the formal definition of constraint grammar G for the head of GEP-rule gene is given as follows:

$$G = (\{E_L, E_{LR}, E_R, E_A, E_T\}, \{l, r, a, v, c\}, E_L, P'),$$

where l, r, a and c are the terminal symbols, and $l \in L$, $r \in R$, $a \in A$, $v \in A_s$, $c \in C_s$. The production set P' is following:

$E_L \rightarrow l\ E_{LR}^{\ t} \mid r\ E_{LR}^{\ t}$, t is the arity of l or r

$E_{LR} \rightarrow E_R \mid E_L$

$E_R \rightarrow r\ E_A^{\ 2} \mid r\ E_A\ E_T \mid r\ v\ E_A \mid r\ v\ E_T$

$E_A \rightarrow a\ v\ E_T^{\ n-1}$, n is the arity of a

$E_T \rightarrow v \mid c$

Set $L_{ang}(G) = \{w \in \{l, r, a, v, c\}^*: E_L \Rightarrow w\}$ is the GEP-rule individual set.

The GEP-rule individuals consist of several attribute-judgment conditions linked with "and" or "or" operations [18]. In this paper, we represent the detectors based on the GEP-rule, and give the following definition for an available detector.

***Definition* 2:** Assuming that $I_{detector}$ consists of m attribute-judgment conditions, (v_i, f_i, b_i) denotes the i-th condition (i =1, … , m), v_i=index(a_i), $a_i \in A_s$, $f_i \in F_s$ and $b_i \in C_s$. then

$$I_{detector} = \{(v_1, f_1, b_1), (v_2, f_2, b_2), \ldots, (v_m, f_m, b_m)\}.$$

In $I_{detector}$, m cannot be larger than the number of attribute genes in an antigen. However, different attribute-judgment conditions can use the same attribute ID.

A GEP-rule individual, its corresponding GEP-rule and detector are all shown in Example3.

- Example3

GEP individual: and. and. ≥. >. <. a23. 10. a5. 3. a5. 300. a21

GEP-rule: (a5>3) and (a5<300) and (a23 ≥ 10)

detector: {(a5,>,3), (a5,<,300), (a23, ≥,10)}

The antigen given in Example1 cannot be detected by the above detector because the value of its 23th attribute (a23) is 1 that isn't equal to or larger than 10, whereas the antigen given in Example2, which satisfies all attribute-judgment conditions of the detector, can be recognized.

4.2 Avidity Function Based on Attribute Priority

Since the existing affinity functions are not adaptable to quantify the interaction between the GEP-presented immune entities, in this subsection, we give definitions of affinity and avidity

in IIDS according to the difference between affinity and avidity in immunology, and distinguish them in terms of numerical values.

(1). Affinity

When a detector matches an antigen, the result is a Boolean value: false or true. Therefore, it is simple to decide their affinity based on only two values**.** If the attribute priority is considered when deciding whether each condition is matched, we can obtain a match degree as the affinity so as to realize not only the regular binary detector match, but also complete scale match and relationship match.

The attribute weight, said w, describes the importance of each characteristic attribute. The value of w can be a given constant or a variable which changes adaptively with the different task requirements.

***Definition* 3:** We use w_{bi} to denote the i-th attribute weight of detector b, which has an alphabet of cardinality 3 with values: t_1, t_2 and t_3.

$$w_{bi} = \begin{cases} t_1 & ; b \text{ is crucial characteristic attribute} \\ t_2 & ; b \text{ is important characteristic attribute} \\ t_3 & ; b \text{ is assist attribute} \end{cases}$$

Based on distance function, the match degree between a detector and an antigen reflects the similarity between them. Based on Definitions 1 through 3, and supposing that there are m judge conditions for a detector, **Table 1** summarizes various notations and the relationship between them.

Table 1. Relationship between detector and antigen

attribute ID		v_1	v_2	…	v_i	…	v_m
detector b	**weight**	w_{b1}	w_{b2}	…	w_{bi}	…	w_{bm}
	function symbol	f_1	f_2	…	f_i	…	f_m
	constant	b_1	b_2	…	b_i	…	b_m
antigen g	**attribute value**	g_1	g_2	…	g_i	…	g_m

In addition, we define a normalized match degree between detector b and antigen g, denoted by M_{bg} as follows. To ensure the smaller distance reflects a better matching, we assume that $t_1 < t_2 < t_3$, and let $max(w)$ denote the maximum value of w.

$$M_{bg}(w) = \frac{\sum_{i=1}^{m} w_{bi}}{m \times max(w)}. \tag{2}$$

Based on the normalized match degree, we use the following definition of the affinity that to describe the strength of interaction between a detector and an antigen.

Definition* 4: *affinity, is 0 when a detector does not match an antigen; otherwise, it replies on their match degree. The affinity between detector b and antigen g is calculated as in (3).

$$affinity(b, g) = \begin{cases} 0 & ; \text{if none of the attributes can be matched} \\ e^{-M_{bg}} & ; \text{others} \end{cases} \tag{3}$$

(2). Avidity

Different from affinity, avidity is the objective function for the candidate solutions, or the metric for evaluating the adaptation of candidate solutions.

Definition 5: ***avidity***, reflects the interaction of one detector and all antigens, and is classified as self avidity and non-self avidity.

Self avidity is the avidity between a detector and self antigens. Let b be a detector, T_{Nb} the number of self antigens recognized by b, and N the original number of self antigens, we use $avidity_{self}$ to denote the self avidity of b, and calculate it as in (4).

$$avidity_{self}(b) = \frac{T_{Nb}}{N} \tag{4}$$

Non-self avidity is the avidity between a detector and all non-self antigens. Let b be a detector, T_{Pb} be the number of non-self antigens detected by b, and N be the original number of non-self antigens, then $avidity_{non\text{-}self}$ is denoted as the non-self avidity of b and can be calculated as in (5).

$$avidity_{non-self}(b) = \frac{\sum_{i=1}^{N} affinity(b, g_i)}{T_{Pb}} \tag{5}$$

Note that (4), which calculates the detection rate of normal network access with a detector, can be used to evaluate the detector's recognition ability to the normal network access (i.e., self antigens). On the other hand, (5) can be used to evaluate the recognition ability to all attack connects of the detector (i.e., non-self antigens) in term of the average match degree considering the priority of network attributes.

4.3 Immune Algorithm

A critical component in any IIDS is the immune algorithm. Neither an NS algorithm nor an affinity-only-based clonal selection algorithm can realize more than one-type of classification. In other words, they can recognize either a normal access or an abnormal attack, but not both. In an NS algorithm, the affinity of a detector is evaluated only according to its match degree with self-antigens. Meanwhile, the affinity calculation in a clonal algorithm only considers the whole matching status of a detector to the non-self antigens.

In order to overcome the deficiencies of these existing approaches, we propose an improved immune algorithm, named avidity-model based clonal selection (AMCS) based on the avidity model of immunology and CGREA. More specifically, we first discuss the value of avidity in (6) and then describe the proposed algorithm.

(1). Avidity Calculation

A good detector involues a high non-self avidity and a low self avidity. Therefore, we first define two functions: f_{self} and $f_{non\text{-}self}$, where the former is an increasing function of $avidity_{self}$, and the latter is a decreasing function of $avidity_{non\text{-}self}$. Thus, we give following two formulas:

$$f_{self}(b) = \exp\left(\frac{(avidity_{self}(b) - 1) \times 10^k}{(10^k + 1)}\right),$$

$$f_{non-self}(b) = \exp\left(\frac{-avidity_{non-self}(b)}{(10^k + 1)}\right),$$

where b denotes a detector, and k is a balance factor used to balance the degree weights for the self avidity and the non-self avidity within the whole avidity of detectors.

Then, the avidity function is proposed as in (6), and a detector with a lower avidity is better.

$$avidity(b) = f_{self}(b) \times f_{non-self}(b). \tag{6}$$

There are two cases that would happen if we use (6) to evaluate a detector individually:

- If $avidity_{self}$ is high while $avidity_{non\text{-}self}$ is low, the value of avidity is larger.
- If $avidity_{self}$ is low but a$vidity_{non\text{-}self}$ is high, the value of avidity is smaller.

In the first case, the detector has a stronger recognition ability to the self antigens, but a weaker recognition ability to the non-self ones, then, NS is applied and the detector is discarded without being used for intrusion detection. In the second case, the recognition ability of the detector to the non-self antigens is stronger, and that to the self ones is weaker, so positive selection is executed, and the detector is reserved as a survival. Therefore, through changing the balance factor k, we can determine which selection process, NS or positive selection, plays the main role in detector evaluation.

(2). Algorithm Description

The basic idea of the proposed AMCS algorithm is as follows. Initially, all detectors satisfying the constraint grammar G are included. The detectors with the lowest avidity in the historical sets are included into the optimal detector set.

The following is a list of notations used to describe AMCS.

$S_{antigen}$: the antigen set;

$S_{C\_detector}$, $S_{best\_detector}$: the clone detector population and the optimal detector set, and their sizes are p_n and p_{best}, respectively, where $p_{best} \leq p_n$;

g_e, g_m, g_{max}: the current evolution generation, the number of continuous non-upgraded generation, the maximum evolutional generation, respectively;

p_1, p_2: the detectors selected for evolution operators;

o_1, o_2: the offspring of detectors $p_1$and p_2, which are generated via evolution.

Before executing the algorithm, we extract the antigens from the set of original dat, such as network packets, log data, etc. These antigens, including self and non-self ones, are stored in set $S_{antigen}$. The description of the AMCS algorithm is given as follows.

Algorithm: avidity-model based clonal selection (AMCS)

Input: $S_{antigen}$

Output: $S_{best\_detector}$

Step1: Generate the initial individuals of $S_{C\_detector}$ (the same as Step1 of CGREA);

Step2: Build $S_{best\_detector}$ and a temporary detector population $S_{tmp\_detector}$ with population size p_n ;

Step3: Submit each antigen of $S_{antigen}$ to $S_{C\_detector}$, calculate the $avidity(I_{detector\_i})$ according to (6), where $\forall\ I_{detector\_i} \in S_{C\_detector}$, i=1, …, p_n, and update $S_{best\_detector}$ in which the individuals are the best p_{best} ones in ($S_{C\_detector} \cup S_{best\_detector}$).

If $S_{best\_detector}$ has not been upgraded during the g_m generations, then go to Step6;

Step4: Select two individuals, said p_1 and p_2, from $S_{C\_detector}$. Then execute the crossover operation and mutation operation, and generate o_1 and o_2 satisfied with the constraint grammar G;

Step5: Add o_1 and o_2 into $S_{tmp\_detector}$. If the population of $S_{tmp\_detector}$ is smaller than p_n, then go to Step 4; otherwise $S_{C\_detector}$:= $S_{tmp\_detector}$; if the current evolution generation does not above the g_{max}, increase g_e by 1 and go to Step3 ;

Step6: Output $S_{best\_detector}$.

---

(3). Algorithm Analysis

The AMCS algorithm evaluates a detector based on the tradeoff between its binding strength with self antigens and that with non-self antigens. As a result, both NS and positive selection are integrated in AMCS. The detectors matching self antigens are eliminated and the

ones binding non-self survive. Consequently, the false alarms for self antigens are reduced, and more non-self antigens are recognized efficiently. How to add new optimal detectors into the optimal set relies on their avidity value in ascending order. Finally, the algorithm assigns the optimal set from the last generation according to the approximate optimal matching degree, instead of the highest matching degree.

Moreover, AMCS converges towards the minimum element set of avidity with the probability 1, due to the following reasons: 1) $S_{best\_detector}$ is the limited Markov chain with positive transfer matrix; 2) the individuals in $S_{best\_detector}$ are the optimal ones obtained from the historical generations and will not be eliminated any longer; 3) within a limited number of steps, the non-optimal detectors will be eliminated with the probability 1.

4.4 Pruning Process

In this subsection, we describe how to elimate the redundant detectors generated by GEP-based immune algorithms. A pruning algorithm, named GEP-based detectors pruning (GDP), is proposed after analyzing the redundant detectors generated by CGREA and AMCS.

(1). Detection of redundant detectors

Some intelligent approaches, such as Decision Trees, apply rule(-based) pruning to solve the overfitting problem, and evolve an accurate and compact rule set. In view of binary classification tasks, the rule-pruning process can be processed in the two following periods: 1) when generating rules, restrict the tree size based on the minimum description length (MDL) to avoid redundant rules which result in overfitting, and this is so called prepruning, and 2) after all the rules having been generated, the redundant ones are deleted via a pruning algorithm with given pruning data sets. The pruning rules in the period 2 should obide by three basic principles:

- Select the simplest rule when there is coverage-overlap according to Occam's razor which is a widely used principle in decision tree learning [26];
- The fewer attributes in a rule is better, as it is easy to be understood and has a lower cost in storing and detecting;
- Find the balance between the number of attributes and the detection rate of rules, and try to build simpler rules on the condition of ensuring detection rate.

Compared to the detector sets obtained from other immune algorithms, each GEP-based detector (i.e. rule) generated by CGREA only contains two or three attributes, and the above principles are almost satisfied before pruning. Therefore, in view of the detector set generated by AMCS based on CGREA, the key points are optimizing the detector sets, deleting redundant detectors, reducing the size of detector set, reducing the memory space, and increasing the detection efficiency.

(2). Algorithm Description

In view of the characteristics of the GEP-based detectors, we propose the GDP algorithm for GEP-IIDS by improving the decision tree pruning algorithm proposed in [27].

In order to describe GDP distinctly, a list of notations is first given as follows.

S_{best_Ab}: the optimal detector set whose initial size is p_n;

$S_{Ag\_pruning}$: the pruning antigen set with the size g_n, which consists of the antigens selected randomly and the antigens used to generate S_{best_Ab};

S_{Ab_pruned} : the mature detector set.

Before executing GDP, S_{best_Ab} has been ordered depending on the avidities of detectors. The GDP algorithm works as follows.

Algorithm: GEP-based Detectors Pruning (GDP)

Input: $S_{Ag\_pruning}$
S_{best_Ab}
Output: S_{Ab_pruned}
Step1: Set $S_{Ab\_pruned} = \varnothing$ and i=1;
Step2: Get the i-th detector b in the current S_{best_Ab} and add it to S_{Ab_pruned};
Step3: Delete all the antigens that match with b;
Step4: If $S_{Ag\_pruning}$ is null, go to Step6;
Step5: $i=i+1$, if $i \le p_n$, go to Step2;
Step6: Output S_{Ab_pruned}.

(3). Algorithm analysis

Without calculating the detectors' avidities repeatedly, GDP can reduce the temporal cost of pruning compared to the original pruning algorithm. The selected order of a detector is only decided according to its avidity calculated based on the given training set. The spatial cost of GDP is the same as that of the original pruning algorithm, but the time complexity is reduced by an order of magnitude. It is very beneficial to IIDS when handling a huge amount of network data. The recalculation of avidities in the original pruning algorithm can be seen as a retraining process that treats the optimal detector set as the original detector set. Nevertheless, efficient pruning redundant detectors using GDP, is based on the premise that the optimal detectors can be generated by using the training set. If the premise cannot be guaranteed, the original pruning algorithm is more feasible than GDP.

Furthermore, GDP is used to delete redundant detectors after the detector set has been generated via AMCS, which is different from the NS algorithm where the detector set is optimized during the detectors generation process. GDP mainly implements the pruning function for the GEP-based detectors which satisfy two properties: closure and integrality. In our previous work [18], the number of optimal detectors generated by CGREA, on which AMCS is based, is small and simple. Therefore, it is reasonable for GDP to judge whether a detector is redundant or not just after generating the optimal detector set.

5. Experimental Results and Analysis

In this section, we describe the experimental environment and parameter settings, and evaluate the proposed GEP-IIDS framework using the well known KDD'99 set.

5.1 Environment and Parameter Setting

The KDD'99 set has been recently utilized extensively for intrusion detection research and system development through a suite of pattern recognition and bio-inspired computing algorithms. It provides two data sets, namely 10% Training Set (kddcup.data_10_percent) and Test Set (corrected) [25]. Any network connect record in these data sets contains 41 features, in which features no.2, 3, 4 are symbolic, and the others are numerical. In our experiments, the symbolic features have also been converted to be numerical, and then all numerical 41 features are normalized. Each record is labeled as either normal, or an attack. The attack records are grouped into one of the four categories: Probing, DoS, User to Root (U2R) and Remote to Local (R2L). The actual numbers of records in the training/test data sets used in experiments

are listed in **Table 2**. The records in the two subsets, namely Training Subset and Test Subset, are randomly sampled from the 10% Training Set.

Table 2. Distribution of the data sets used for training and test

Category	10% Training Set	Training Subset	Test Set	Test Subset
Normal	97278	986	60593	4000
Probing	4107	41	4166	1107
DoS	391458	3961	229853	13715
U2R	52	1	228	52
R2L	1126	11	16189	1126
total	494021	5000	311029	20000

All experiments were performed on an Intel-Pentium® D CPU 2.8GHz platform with 1GB RAM size running Windows XP. The performance of intrusion detection is evaluated in terms of detection rate p_d and false alarm rate p_f using the following equations:

$$p_d = \frac{\text{the number of detected attacks records}}{\text{the total number of attacks records}},$$
$$p_f = \frac{\text{the number of Normal records detected as attacks}}{\text{the total number of Normal records}}. \quad (7)$$

(1- p_d) is equal to the probability of false negative (i.e. the attack records are undetected), and p_f denotes the probability of false positive (i.e., the normal records are mistaken for attack ones).

In our implementation of the proposed GEP-IIDS framework, each detector is a single-gene GEP individual in which the function set F_s is {and, or, not, >, ≥, <, ≤, =, !=}. In addition, each element of constant set C_s is a random constant∈[0.1], and the attribute set A_s is { $a1$, $a2$, … , $a41$} corresponding to the feature set for the KDD'99 records. The other GEP parameters are set as follows: crossover probability p_{ross} =0.2, mutation probability p_{mute} =0.95, length of head h=8, function symbol probability p_{func} =0.6, attribute symbol probability p_{attr} =0.28, and roulette wheel selection is conducted. To calculate the attribute weight, we set t_1=1, t_2=2 and t_3=3. According to the preliminary results, we set k=2 and p_{best} =20 in the following experiments.

5.2 Results and Analysis

(1). AMCS vs. CGREA

The proposed AMCS algorithm is executed to generate detector set using the Training Subset as training set, and tested on the 10% Training Set and the Test Subset **Table 2**. **Fig. 2** summarizes the performance of 12 training runs in terms of detection rate and false alarm rate. The false alarm rates from the 24 detection tests (each of the 12 detector sets is tested on two test sets) range from 0.1% to 0.5%. Tested on the 10% Training Set, the detection rates are mostly between 98.8% ~ 97.6%, which are much better than those of the Test Subset (90.5%~88.4%). The result in **Fig. 2** indicates that the optimal detector sets generated by AMCS can achieve a lower false alarm rate and a consistent detection rate. If the distribution of training set is the same as that of test set, the detection performance of the AMCS algorithm will be better.

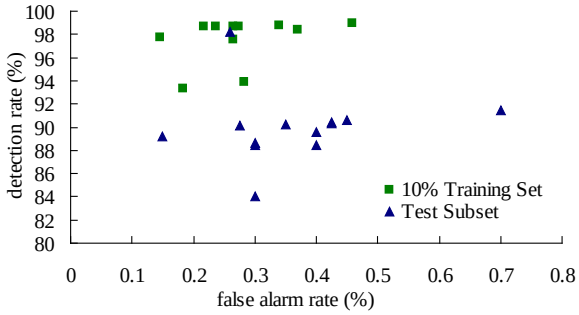
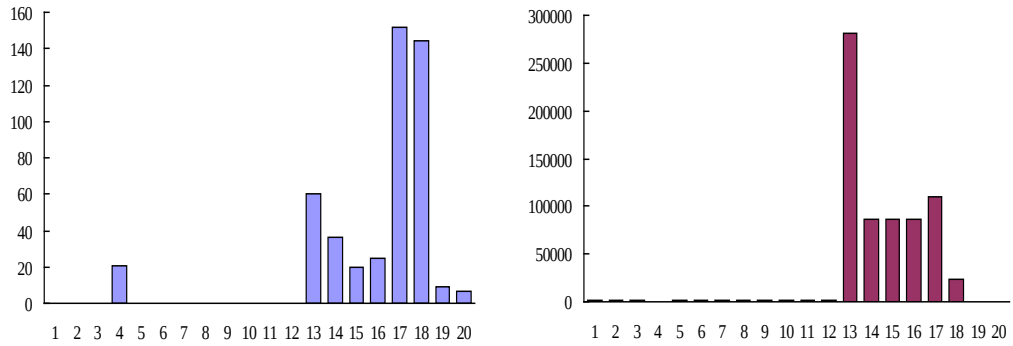
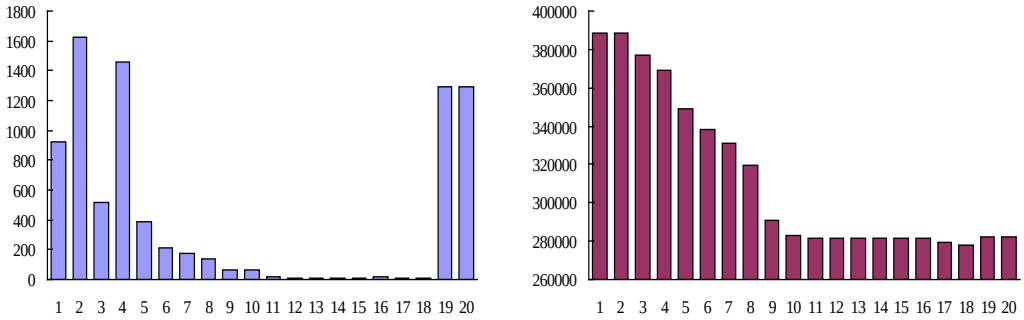


Fig. 2. AMCS algorithm performance of 12 runs

Testing on the 10% Training Set, the coverage of a single detector in the optimal detector sets, which is generated by AMCS and CGREA respectively, are compared in **Fig. 3**. The larger number of covered Normal records denotes more false negative errors, and the more covered attack records indicate the higher detection probability of a given detector. The detectors in the optimal sets, extracted by using the two methods, are both ranked on the avidity values. The detector, with a smaller value, is better and given a smaller serial number.



(a) Number of Normal records covered by AMCS detector **(b)** Number of attack records covered by AMCS detector



(c) Number of Normal records covered by CGREA detector **(d)** Number of attack records covered by CGREA detector

X-coordinate is the serial number of a detector in the optimal detector set, and Y-coordinate is the number of records covered by the detector.

Fig. 3. Coverage of each detector in the optimal detector set

Fig. 3 (a) and **(b)** show000 that, in AMCS, except for the fact that detector no.4 covers only 128 Normal records, none of Normal records and only a very small number (only 970~1238) of attack records are covered by the detectors no.1 through no.12. In addition, even through detectors no.14 ~ no.16 cover a fewer Normal records, detector no.13 has a higher avidity due to the fact that it covers three times of the attack records covered by others. It is the same case in detectors no.17~ no.20. In AMCS, the avidity function is based on attribute-priority, and uses the balance factor k to integrate two selection processes, which is valid for evaluating detectors. The value of k in our experiments is 2. It means that the power of self avidity weight is twice of that of non-self avidity, and negative selection prevails in the evolution process. Therefore, the AMCS-based detectors which cover less Normal records are more optimal.

Because CGREA evaluates detectors considering the coverage probabilities of attacks and the approximate degree of distributions of the training set and the covered set, it can be seen from **Fig. 3 (c)** and **(d)** that the CGREA-based detector covers more attack records and more Normal records as well, thus resulting in a lower avidity. We denote by N_c the sum of the numbers of attack records covered by all detectors in a detector set. According to the results given in **Fig. 3 (c)** and **(d)**, the N_c of AMCS-based and CGREA-based optimal detector set are 687774 and 6247007 respectively, and both are much larger than 396743 (the total number of attack records in the 10% Training Set). It is obvious that the overlapping coverage is also a serious problem of the GEP-based detector sets.

As we can see from **Fig. 3**, the results show that the diversity of the AMCS-based detector population is more sufficient, and the degree of coverage-overlap between detectors is lower than that of CGREA-based detectors population. Furthermore, the proposed avidity function, which is capable to balance the weight of NS and positive selection in AMCS via balance factor, is more applicable to IIDS than that of CGREA.

(2). Performance evaluation of the GDP algorithm

In order to evaluate the sensitivity of the proposed GDP to the pruning test data, we randomly generate five subsets from the 10% Training Set with different sizes and different record distribution as pruning antigen sets. These five pruning antigen sets are outlined in **Table 3**, and p1 is also used as the training set to generate the optimal detector set before pruning. The distribution of p1, p2 and p3 is almost the same with the 10% Training set. The distribution ratios in p4 and p5 are very different with those in the 10% Training Set and KDD test set.

Selecting p5 as the pruning antigen set and the 10% Training set as the detection test set,

Table 3. Distribution of the pruning subsets

Pruning antigen set	**Record number**	**Record distribution (%)**				
		Normal	Probing	DoS	R2L	U2R
p1	5000	19.70	0.82	79.62	0.22	0.02
p2	10000	19.70	0.82	79.24	0.23	0.01
p3	20000	19.70	0.83	79.23	0.23	0.01
p4	20000	20.00	5.54	68.58	5.63	0.26
p5	30000	33.33	13.69	49.05	3.75	0.17

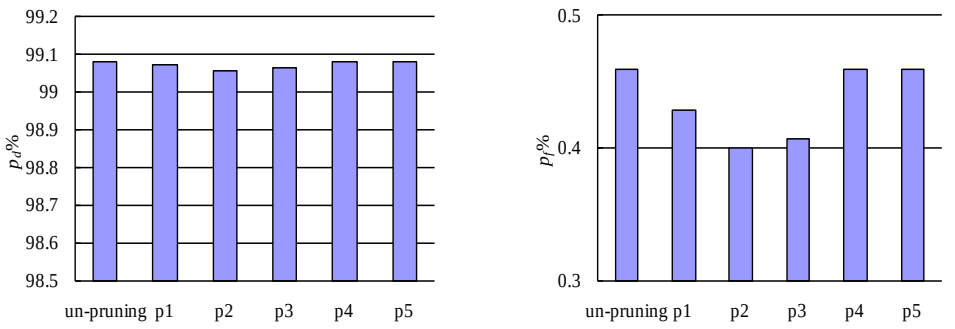
As the records are randomly selected according to given ratio, the accuracy of the given ratio may result in some subtle differences in the number of records with different type between these subsets.

Table 4. Comparation between the original and pruned optimal detector sets

Scheme	Original optimal detector set			Pruned optimal detector set		
	size	number of detected Normal records	number of detected attack records	size	number of detected Normal records	number of detected attack records
AMCS	20	447	392648	9	447	392648
CGREA	20	3070	392146	3	3069	392144

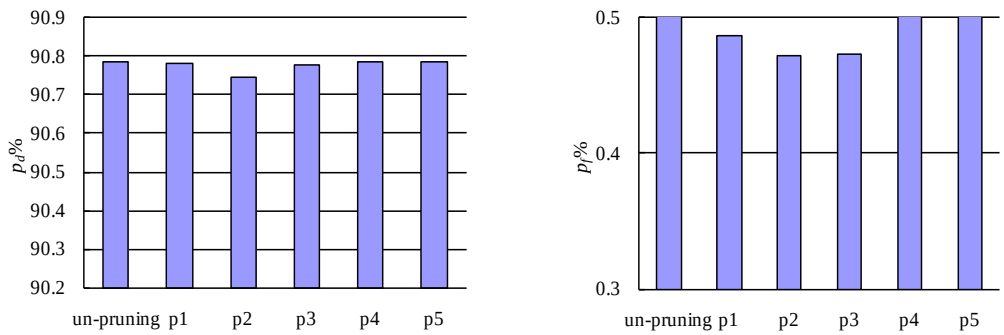
The test results of the detector sets before and after being pruned are both listed in **Table 4**. The number of detected records detected by the pruned optimal detector set is the same as that of original one generated by AMCS. The number of detected attack records based on CGREA is reduced by two, and the number of Normal records mistaken for attack is reduced by one. However, the reduction is insignificant compared to the total number of attack records (396743) and that of Normal records (97278). It is worth noting that, in these two cases, the sizes of pruned optimal detector sets are only 15% and 45% of the original ones while achieving almost the same detection performances. The result in **Table 4** shows that GDP can improve the detection efficiency of the IIDS through pruning the optimal detector set.

The detection performance of pruned detector set is presented in **Fig. 4** and **Fig. 5**. We use the five sets given in **Table 3** as the pruning antigen sets, then detect on the 10% Training Set and the Test Set using the AMCS-based detector set before and after being pruned, respectively. In each case, the detection rate and false alarm rate both differ by no more than 0.1%. It indicates that GDP can achieve good detection performance constantly.



(a) Detection rate **(b)** False alarm rate

Fig. 4. Performacne of detecting the 10% Training Set



(a) Detection rate **(b)** False alarm rate

Fig. 5. Performacne of detecting the Test Set

Next we will analyze the subtle differences after different sets are pruned.

- When the pruning antigen set is p1, which is also the training set for generating the optimal detector set, the detection rate is reduced by 0.08% and 0.05% comparing to those of un-pruned ones, but the false alarm rate is reduced by 3.1% and 2%.
- When the pruning antigen sets are p2 and p3 whose sizes are increased to one and three times more than the training set, respectively, the achived detection rate is higher than those of the pruning using p1 and the unpruning case, and the false alarm rate is reduced significantly.
- When the pruning antigen set are p4 and p5, whose sizes are increased to four and six times of the training set respectively, the detection performance are the same with those of being pruned.

The results in **Fig. 4** and **Fig. 5** show that it is viable to treat the original training set as the pruning antigen set, but it will be better when the size of pruning antigen set is larger than that of the original training set. However, if the size of pruning antigen set exceeds a threshold, the detection performance will no longer improve, even though the time spent in pruning increases.

(3). Performance comparison

Table 5 presents a comparison of detection performance with some other intelligent methods using the Test Set for test. In GEP-IIDS, the one that provides the best detection performance among the twelve sets in **Fig. 2**, is selected as the detector set. As can be seen from **Table 5**, although the attack detection rate achieved by GEP-IIDS is slightly lower than that achieved by most other methods, but the false alarm rate is close to that of KDD'99 winner 1. The highest attack detection rate is obtained by using Neural Networks with a much higher false alarm rate (2.13%) than 0.51% obtained with GEP-IIDS. Using the GEP-IIDS detectors, DoS records are detected with a probability of 97.26%, that is only slightly lower than that of KDD'99 winner 2 but higher than others. To state that there are other results in the literatures that even overcome the KDD'99 winners, but they have been obtained by using filtered versions of the KDD'99 set. In order to have a fair comparison, we have not included them.

In order to detect the 22 types of attack in the 10% Training Set and achieve the performance presented in **Table 5**, GEP-IIDS merely used nine (the size of the pruned optimal detector set) bi-attribute detectors. In comparison, other methods consume more time and space resources. For instance, KDD winner 1 contained 500 decision trees for 22 specific attack types [28], while KDD winner 2 used 218 and 755 decision trees for five categories and specific attack types respectively [29]. In addition, Neural Networks assigned 125 input neurons and five output neurons, and another method based on Decision Trees generated 218 and 537 decision trees for five categories and specific attack types respectively in [30]. The system based on GP-classifier generated a set of 50 optimal transformations to achive the performance [31].

Table 5. Comparison with other intelligent methods

Method	Attacks detection rate(%)	Attacks false alarm rate(%)	DoS detection rate(%)
GEP-IIDS	90.67	0.51	97.26
KDD winner 1[28]	91.00	0.50	97.10
KDD winner 2 [29]	91.30	0.58	97.47

Neural Networks [30]	92.61	2.13	97.00
Decision Trees [30]	91.94	0.57	97.00
GP-classifier [31]	92.50	1.35	96.00

The above discussion indicates that the GEP-IIDS can generate optimal detector set which can obtain a low false alarm rate and a high DoS attack detection rate with less time in generating detectors and detecting attack/intrusion.

We have also compared GEP-IIDS with *v*-detectors, one of the main and best NS algorithms. We choose the *v*-detectors NS algorithm because it also used the same 10% Training Set for evaluation. Since the results of immune algorithms like these are stochastic and mostly depend on the parameter settings, we use the best outcome of the *v*-detectors NS algorithm presented in the literatures for comparision, and the results are given in **Table 6**. They indicate that the propsed GEP-IIDS can achieve a better performance, even with a small population size and a smaller maximum evolutional generation which translates to a lower space and time complexity.

Table 6. GEP-IIDS vs. *v*-detectors NS

Option	**GEP-IIDS**	***v-detectors* NS [22]**
size of clone detector population	60	1000
maximum evolutional generation	50	1000
size of pruned optimal detector set	9	100~800
detection rate (%)	98.97	83.92
false alarm rate (%)	0.46	1.45

6. Conclusions

While the poor performance of prior approaches based on immune-inspired IDS has led to concerns about the viability of such approaches, we believe that immune-inspired IDS is still promising as it considers not only how to generate the detectors/rules for intrusion detection, but also how to build a complete security system for networks. In this paper, we have proposed and demonstrated the applicability of an improved framework GEP-IIDS integrating two bio-inspired techniques: AIS and GEP, to overcome two major problems - scalability and coverage - of existing IIDSes. Our main contributions are: 1) an attribute-gene representation for antigens and a GEP-rule based representation for detectors, 2) new avidity/affinity functions that take weighting attribute priorities into consideration, 3) an avidity-model based clonal selection algorithm, which integrates both negative selection and positive selection, and 4) a GEP-based detectors pruning algorithm for GEP-IIDS to find and eliminate the redundant detectors. The experiment results have shown that the GEP-IIDS framework provides a higher detection probability of DoS attack, a lower false alarm rate and a lower optimal detectors generation cost. Moreover, our solution requires much less computing resources during the detection procedure since the size of the optimal detector set in use is very small and each detector needs only two attributes. As shown in our study, it is feasible and effective to eliminate the bottlenecks of immune-inspired intrusion detection using efficient approaches to adjusting basic elements and pruning redundant detectors.

References

[1] L. Zhou, B. Zheng, A. Wei, B. Geller and J. Cui, “A Scalable Information Security Technique: Joint Authentication-Coding Mechanism for Multimedia over Heterogeneous Wireless

Networks," *Wireless Personal Communications*, vol. 51, no. 1, pp. 5-16, Oct. 2009. Article (CrossRef Link)
[2] L. Zhou, A. Vasilakos, N. Xiong, Y. Zhang and S. Lian, "Scheduling Security-Critical Multimedia Applications in Heterogeneous Networks," to appear in *Computer Communications*. doi:10.1016/j.comcom.2010.01.009
[3] K. Butler, T. R. Farley, P. McDaniel and J. Rexford, "A survey of BGP Security Issues and Solutions," in *Proc. of the IEEE*, vol. 98, no. 1, pp.100-122, Jan. 2010. Article (CrossRef Link)
[4] P. Owezarski, "On the impact of DoS attacks on Internet traffic characteristics and QoS," in *Proc. of 14th International Conf. on Computer Communications and Networks (ICCCN'05)*, San Diego, California USA, pp.269-274, Oct. 2005. Article (CrossRef Link)
[5] T. Peng, C. Leckie and K. Ramamohanarao, "Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems," *ACM Computing Surveys*, vol. 39, no. 1, article 3, pp.1-42, Apr. 2007. Article (CrossRef Link)
[6] J. Kim, P. Bentley, U. Aickelin, J. Greensmith, G. Tedesco and J. Twycross, "Immune system approaches to intrusion detection - a review," *Natural Computing*, vol. 6, no. 4, pp. 413-466, Dec. 2007. Article (CrossRef Link)
[7] D. Dasgupta, "Advances in artificial immune systems," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 40-49, Nov. 2006. Article (CrossRef Link)
[8] S. A. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," *Evolutionary Computation*, vol. 8, no. 4, pp. 443-473, Dec. 2000. Article (CrossRef Link)
[9] Z. Ji and D. Dasgupta, "Augmented negative selection algorithm with variable-coverage detectors," *in Proc. of Congress on Evolutionary Computation (CEC'04)*, Portland, Oregon, USA, pp. 1081-1088, June 2004. Article (CrossRef Link)
[10] Z. Ji, "Negative selection algorithms: from the thymus to *v*-detectors," *University of Menphis, USA, Ph. D thesis*, 2006.
[11] J. Kim and P. Bentley, "Immune memory and gene library evolution in the dynamical clonal selection algorithm," *Journal of Genetic Programming and Evolvable Machines*, vol. 5, no. 4, pp. 361-391, Sep. 2004. Article (CrossRef Link)
[12] J. Kim, "Integrating artificial immune algorithms for intrusion detection," *University of London, UK, Ph. D thesis*, 2002. Article (CrossRef Link)
[13] E. Hart and J. Timmis, "Application areas of AIS: The past, the present and the future," *Applied Soft Computing*, vol. 8, no. 1, pp. 191-201, Jan. 2008. Article (CrossRef Link)
[14] T. Stibor, "On the appropriateness of negative selection for anomaly detection and network intrusion detection," *Darmstadt University of technology, Germany, Ph. D thesis*, 2006. Article (CrossRef Link)
[15] J. Jimmis. R. d. Lemos. M. Ayara and R. Duncan. "Towards immune inspired fault tolerance in embedded," in *Proc. of the 9th International Conf. on Neural Information Processing (ICONIP'02)*, Orchid Country Club, Singapore, University of Kent at Canterbury Printing Unit, pp.1459-1463, Nov. 2002. Article (CrossRef Link)
[16] T. Stibor, J. Timmis and C. Eckert, "On the appropriateness of negative selection defined over hamming shape-space as a network intrusion detection system," in *Proc. of the Congress on Evolutionary Computation (CEC'05)*, Edinburgh, UK, IEEE Press., pp. 995-1002, July 2005. Article (CrossRef Link)
[17] C. Ferreira, "Gene expression programming: Mathematical modeling by an artificial intelligence," 2nd ed. Springer-Verlag, Germany, 2006. Article (CrossRef Link)
[18] J. Skaruz, "Detecting Web application attacks with use of gep expression program," in *Proc. of the Eleventh conf. on Congress on Evolutionary Computation (CEC'09)*, Trondheim, Norway, IEEE Press, pp. 2029-2035, May, 2009.
[19] W. Tang, Y. Cao, X. M. Yang and W. H. So, "Study on adaptive intrusion detection engine based on gene expression programming rules," in *Proc. of International Conf. on Computer Science and Software Engineering (CSSE'08*), Wuhan, China, pp.959-963, Dec. 2008. Article (CrossRef Link)
[20] U. Aickelin, "Artificial immune system," *Introductory tutorials in optimization, decision support and search methodology*, Chapter 13, Kluwer, 2005.

[21] M. Ayara, J. Timmis, L.N. d Lemos and R. Duncan, "Negative selection: how to generate detectors," in *Proc. of the 1st International Conf. on Artificial Immune Systems (CARIS '02)*, University of Kenta at Canterbury, pp. 89-98, Sept. 2002. Article (CrossRef Link)
[22] S. Mariathasan, R. G. Jones and P. S. Ohashi, "Signals involved in thymocyte positive and negative selection," *Seminars in Immunology*. vol. 11, pp. 263-272, Aug. 1999. Article (CrossRef Link)
[23] Z. Ji and D. Dasgupta, "Applicability issues of the real-valued negative selection algorithms," in *Proc. of IEEE Congress on Evolutionary Computation Conference (CEC'03)*, Canberra, Australia, ACM, pp. 111-118, Dec. 2003. Article (CrossRef Link)
[24] L. N. de Castro and F. J. V. Zuben, "The clonal selection algorithm with engineering applications," in *Proc. of GECCO'00*, Las Vegas, Nevada, USA, pp.36-39, July, 2000. Article (CrossRef Link)
[25] F. Liu and L Luo, "Immune clonal selection wavelet network based intrusion detection," in *Proc. of Artificial Neural Networks: Biological Inspirations-ICANN 2005*, LNCS, vol. 3696, Springer, pp. 331-336, 2005. Article (CrossRef Link)
[26] G.F. Luger, "Artificial intelligence: structures and strategies for complex problem solving," 6th Ed., England: Addision Wesley, 2008. Article (CrossRef Link)
[27] KDD CUP'99 DATA Set, http://kdd.ics.uci.edu/data bases/kddcup99/kddcup99.html
[28] C. Zhou, W. Xiao and T.M. Tirpak, "Evolving accurate and compact classification rules with gene expression programming," *IEEE Transaction on Evolutionary Computation*, vol. 7, no. 6, pp. 519-531, Dec. 2003. Article (CrossRef Link)
[29] C. Elkan, "Results of the KDD'99 Classifier Learning," *ACM SIGKDD 2000*, Boston, MA, USA, vol. 1, no. 2, pp. 63-64, Aug. 2000.
[30] I. Levin, "KDD99 classifier learning contest LLsoft's results overview," *ACM SIGKDD 2000*, Boston, MA, USA, vol. 1, no. 2, pp. 67-75, Aug. 2000. Article (CrossRef Link)
[31] Y. Bouzida and F. Cuppens, "Neural networks vs. decision trees for intrusion detection," in *Proc. of IEEE / IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM)*, Tuebingen, Germany, pp. 81-88, Sep. 2006. Article (CrossRef Link)
[32] K. Faraun and A. Boukelif, "Genetic programming approach for multi-category pattern classification applied to network intrusions detection," *International Arab Journal of Information Technology*, vol. 4, no. 3, pp. 237-246, July 2007. Article (CrossRef Link)

Wan Tang received the B.S. and M.S. degrees in Computer Application Technology from South-Central University for Nationalities (SCUN), Wuhan, China, in 1995 and 2001, respectively, and received the Ph.D. degree in Communication and Information System from Wuhan University, China in 2009. She is currently an associate professor in the School of Computer Science of SCUN. Also, from 2001 to 2002, she worked as a visiting researcher at the Advanced Communications and Networks Laboratory at Chonbuk National University, Jeonju, South Korea. Her research interests include protocols for optical/wireless communication networks, network security, and computational intelligence.

Limei Peng is currently working as a post-doctorate fellow in Grid Middleware Research Center, Korea Advanced Institute of Science and Technology in South Korea. She received her B.S degree from the South-Central University for Nationalities in Wuhan, China, in 2004, and her M.S. and Ph.D degrees from the Chonbuk National University in Jeonju, Chonbuk, South Korea, in 2006 and 2010, respectively. Her research interests include network architectures, and protocols for optical communication networks, optical fiber sensor networks, and Grid networks.

Ximin Yang received the B.S. and M.S. degrees in Computer Application Technology from South-Central University for Nationalities (SCUN), China, in 1994 and 2003, respectively. He is currently an associate professor in the School of Computer Science of SCUN. His research interests include network storage system and information security.

Xia Xie is currently an associate professor in the school of Computer Science & Technology of Huazhong University of Science and Technology (HUST), China. She received her Ph.D. degree from HUST in 2006. Her research field includes performance evaluation, grid, and high performance computing.

Yang Cao is a professor in the School of Electronic Information at Wuhan University, and Chief of the National IC Design Shenzhen Base–Wuhan University Joint Laboratory for EDA. He worked as a visiting scholar at Technische Universität München, Universität Hamburg, and Siemens Research Center, Germany during the period between 1984 and 1998. He was invited to the Department of Electronic & Computer Engineering, Hong Kong University of Science and Technology in 2003. His current research interests include next generation network security, wireless sensor networks, and SoC design methodology and technology.