Improving the Generalization Capability of HIDMA with DeJong's Gene Expression

Jungan Chen, Feng Liang, Zhaoxi Fang
Electronic Information Department
Zhejiang Wanli University
No.8 South Qian Hu Road
Ningbo, Zhejiang, 315100, China
friendcen21@hotmail.com, liangf_hz@hotmail.com, zhaoxifang@gmail.com

Abstract: - In this work, an augmented hybrid immune detector maturation algorithm applied in anomaly detection is proposed. In order to improve the generalization capability, the DeJong's gene expression is used. Experiment results show the algorithm is more effective than other algorithms with binary string expression.

Key-Words: - Artificial immune system, generalization capability, hybrid immune detector

1 Introduction

Nowadays, Artificial Immune System (AIS) has been applied to many areas such as computer security, classification, learning and optimization [1]. Negative Selection Algorithm, Clonal Selection Algorithm, Immune Network Algorithm and Danger Theory Algorithm are the main algorithms in AIS [2][3].

Reference [4] mentioned 'Th-cells are general, nonspecific detectors, and so are not efficient at detecting specific pathogens. B-cells, by contrast, can adapt to become more specific, and thus more effective at detecting particular pathogens'. So there is a balance between generality and specialty. The detectors generated by T-detector Maturation Algorithm (TMA) are just like the generalized lymphocytes (or Th-cells) and the detectors generated by affinity maturation are the specialized lymphocytes (or B-cells). So it is reasonable that Hybrid Immune Detector Maturation Algorithm (HIDMA) combines TMA with affinity maturation. HIDMA with Lifecycle Model (HIDMA-LM) is proposed to solve the adaptive problems. Lifecycle is used as a pressure to improve the effect of the selection operator [5]. To improve the generalization capability and detect common patterns, an augmented HIDMA-LM algorithm called HIDMA-GC (Generalization Capability) is proposed and applied in anomaly detection [6].

In this paper, DeJone's gene expression is used to encode the detector to improve the generalization capability. HIDMA with DeJone's Gene Expression (HIDMA-DJ) is proposed.

2 Algorithm

2.1 DeJong's Gene Expression

Simple gene expression proposed by De Jong, where a fixed-length internal representation for classifier rules is used to express the detector, is used to encode the detectors used in intrusion detection. [7].

Each fixed-length rule will have many feature tests, one for each feature test.The feature test is represented by a fixed-length binary string, the length of which will depend on the type of feature.

Suppose there is a feature test 'F_i' with N_i kinds of feature's type. 'i' is the index value. The feature test can be represented as the following. The value 1 means matched.

$$F_i = \{0,1\}^{N_i} \quad (1)$$

M is the number of the feature tests. The fixed-length rule 'R' can be represented as the following:

$$R = [F_0, F_1, ...F_i, ...F_{M-1}] \quad (2)$$

The antigen is represented by a fixed-length int array as the following, 'i' is the index value referenced to R, g_i is the index value of the feature test F_i.

$$G = [g_0, g_1, ...g_i, ...g_{M-1}] \quad (3)$$

For example, one classifier rule has the two features F1 and F2. In Table.1, if the legal values for the feature Fl are the days of the week, then the

pattern '1000001' would represent the test for Fl being a weekday. Similarly, if the legal values for the feature F2 are the type of network package such as UDP, TCP, HTTP, then the patter 011' would represent the test for F2. The bit 1 in patterns means matched. So the pattern's value [1000001,011] means Monday or Sunday in F1 and TCP or HTTP in F2.

Table.1

F1							F2		
Mon.	Tues.	Wed.	Thurs.	Fri.	Sat.	Sun.	UDP	TCP	HTTP
1	0	0	0	0	0	1	0	1	1

Table.2

F1							F2		
5							2		
0	1	2	3	4	5	6	0	1	2
Mon.	Tues.	Wed.	Thurs.	Fri.	Sat.	Sun.	UDP	TCP	HTTP

Table.3

F1							F2		
5							**2**		
1	0	0	0	0	**0**	1	0	1	**1**
0	1	2	3	4	**5**	6	0	1	**2**
Mon.	Tues.	Wed.	Thurs.	Fri.	**Sat.**	Sun.	UDP	TCP	**HTTP**

As for the expression for an antigen, it will be expressed by using fixed-length int array. Each feature will be expressed by the index value of its type. For example, in Table.2, there is an antigen with pattern '[5,2]' . The value 5 is the sixth type 'sat.' in F1 and 2 is the third type 'HTTP' in F2.

Supose there are a detector encoded with the classifier rule R_j and a antigen G_j. The distance between R_j and G_j is:

$$g_{ji} = G_j[i] \tag{4}$$

$$F_{jji} = R_j[g_{ji}] \tag{5}$$

$$d(G_j, R_j) = \sum_{i=0}^{M-1} F_{jji} \tag{6}$$

For example, in table.3, the distance between the detector '[1000001,011]' and the antigen '[5,2]' will be 1.

2.2 Match Range Model

The detector is defined as dct = {<Sb, selfmin, selfmax> | Sb∈R,selfmin, selfmax∈N}. selfmax is the maximized distance between dct.Sb and selves and selfmin is the minimized distance. The detector set is defined as DCTS. Selfmax and selfmin is calculated by setMatchRange(dct, selves), i ∈ [1, |selves|], $self_i \in selves$:

$$\text{setMatchRange} = \begin{cases} \text{selfmin} = \min(\{d(self_1, dct.Sb)\}) \\ \text{selfmax} = \max(\{d(Self_1, dct.Sb)\}) \end{cases} \tag{7}$$

[selfmin,selfmax] is defined as self area. Others are as nonself area. The antigen is defined as Ag= {< Sg > | Sg∈G}, The antigen set is defined as AGS.

Suppose there is one antigen ag∈AGS and one detector dct ∈ DCTS. When d(ag.Sg,dct.Sb) ∉ [dct.selfmin, dct.selfmax], x is detected as anomaly. It is called as Range Match Rule (RMR) shown in equation 8.Value true means that x is anomaly.

$$\text{RMRMatch}(ag, dct) = \begin{cases} \text{false}, d(ag.Sg, dct.Sb) \in [dct.seflmin, dct.seflmax] \\ \text{true}, d(ag.Sg, dct.Sb) \notin [dct.seflmin, dct.seflmax] \end{cases} \tag{8}$$

Based on RMR, the detect procedure detect(ag,DCTS) is defined as equation 9. True means that x is anomaly.

$$\text{Detect}(ag, \text{DCTS}) = \begin{cases} \text{true}, \exists dct_k \in \text{DCTS}, \text{RMRMatch}(ag, dct_k) = \text{true} \\ \text{false}, \quad \text{others} \end{cases} \tag{9}$$

2.3 The State Transformation Model

In this model, The antigen is redefined as Ag= {<Sg,state,undetectedCount>|Sg ∈ G ,undetectedCount ∈ N, state ∈ {'new','suspect','self','nonself'}}.

Antigen has four states shown in Fig.1. State 'new' means an antigen just inject into the algorithm and 'nonself' means an antigen is detected by the detectors. If an antigen cannot be detected after one generation, its state is changed to 'suspect' and its undetectedCount is inscreased. If an antigen cannot be detected over many generations and its undetectedCount is bigger than the max undetected generations, maxUndetectedCount, its state is

changed to 'self'. The 'self' and 'nonself' will be removed from AGS.

The detector is redefined as dct = {<Sb, d,harmmax, selfmin, selfmax,state,lifecycle, detectedAgNum, oldDetectedAgNum > | d,harmmax, selfmin, selfmax,lifecycle, detectedAgNum, oldDetectedAgNum ∈ N,sate ∈ {'new','highest','maturation','die'}, Sb ∈ R}. Detector has also four states shown in fig.2. State 'new' means a detector or lymphocyte is just generated. If a detector detects any nonself antigen, its state is changed to 'maturation'. If a detector has the highest distance or affinity with a specific antigen than other detectors, its state is changed to 'highest'. Otherwise, its state is changed to 'die' and it will be removed from DCTS.

Other properties in the definition of Ag, dct are calculated by the following steps:

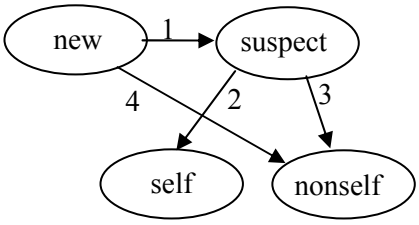
$$M = |AGS|, N = |DCTS|, i \in [1, M], j \in [1, N] \quad (10)$$

The value i is the index of antigen in AGS and the value j is the index of detector in DCTS. The value of d is the distance between dct and current antigen Ag.

1. Equation 11 is used to calculate the distance or affinity between Ag and dct.

2. Equation 12,13 is used to calculate the property of dct's harmmax. The children of the detector dct reproduced are proportion to Harmmax.

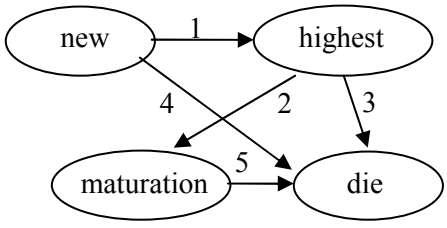
3. According equation 14, if angtigen i cannot detect by any detector in DCTS, antigen i is taken as suspect antigen, which means that antigen i required to be detected over many generations until antigen is is taken as self antigen or nonself antigen. If antigen i can not be detected over maxundetectedCount generations, it is changed to 'self antigen'. Suppose the detector x has the max distance with ag_i, the detector x is changed to highest detector.



1. can not be detected by existed maturation detectors
2. can not be detected over specific generation

3,4. detected by any new or maturation detectors

Fig. 1. antigen's transformation model



1 has the highest affinity with antigen
2 detected an nonself antigen
3,4 has lower affinity with all antigens and can not detect any antigen
5 the lifecycle decreased to 0

Fig. 2.detector's transformation model

$$dct_j.d_i = d_{ij} = d(Ag_i.Sg, dct_j.Sb) \tag{11}$$

$$d_{*j} = \{d_{1j}, d_{2j}, ...d_{Mj}\} \tag{12}$$

$$dct_j.harmmax = \max(d_{*j}) \tag{13}$$

$$\begin{aligned} &if(!detect(Ag_i, DCTS)) \\ &\begin{cases} Ag_i.state = 'suspect' \\ Ag_i.undetectedCount = Ag_i.undetectedCount + 1 \\ dct_x.state = 'highest' \quad , \exists dct_x, dct_x.harmmax = \max(d_{i*}) \end{cases} \\ &if(Ag_i.undetectedCount > maxundetectedCount) \\ &Ag_i.state = 'self' \end{aligned} \tag{14}$$

$$\begin{aligned} &if(RMRMatch(Ag_i, dct_y)) \\ &\begin{cases} Ag_i.state = 'nonself' \\ dct_y.state = 'maturation' \\ dct_y.detectedAgNum = dct_y.detectedAgNum + 1 \\ DCTS_{iM} = DCTS_{iM} \cup dct_y \end{cases} \end{aligned} \tag{15}$$

$$\begin{aligned} &\exists dct_m \in DCTS_{iM}, \quad dct_m.detectedAgNum = \max(dct_*.detectedAgNum) \\ &AgNum = dct_m.detectedAgNum - dct_m.oldDetectedAgNum \\ &if(AgNum > 0)\{ \\ &dct_m.lifecycle = dct_m.lifecycle + \alpha * AgNum \\ &dct_m.oldDetectedAgNum = dct_m.detectedAgNum \\ &\} \end{aligned} \tag{16}$$

$$\begin{aligned} &\exists dct_m \in DCTS, dct_m.lifecycle = dct_m.lifecycle - 1 \\ &if(dct_m.lifecycle == 0), dct_m.state = 'die' \end{aligned} \tag{17}$$

4. In equation 15, if the antigen i is detected by detector y, it is changed to nonself antigen and detector y is changed to maturation detector. The detectedAgNum of detector y is increased 1. DCTSiM is defined as the set of detectors which can detect the antigen i as nonself.

In equation 16, α is a parameter used to control the lifecycle of maturation detector. Suppose detector dctm has the max detectedAgNum, dctm's lifecycle is increased after agi is detected. So dctm can be reserved to detect more common antigen and the generalization capability of the algorithm is improved. If α is set to ∞, it will not die.

5. In equation 17, the lifecycle of the detector will decrease. If lifecycle is equal to 0, it will be removed

2.4 The Detect Process

The algorithm proposed has combined TMA with Affinity Maturation. So it has two detect processes. Some variables are defined in equation 18~21.

$$\forall Ag_{new} \in AGS_{new} \in AGS, Ag_{new}.state = 'new' \tag{18}$$

$$\forall Ag_{suspect} \in AGS_{suspect} \in AGS, Ag_{suspect}.state = 'suspect' \tag{19}$$

$$\forall dct_{highest} \in DCTS_{highest} \in DCTS, dct_{highest}.state = 'highest' \tag{20}$$

$$\forall dct_{maturation} \in DCTS_{marturation} \in DCTS, dct_{highest}.state = 'maturation' \tag{21}$$

1.AGSnew is first detected by DCTSmaturation,called TMADetect. After the process, Angtigen in AGSnew will be split into suspect or nonself antigen. If one antigen cannot detected by any detector, highest detectors will be generated.

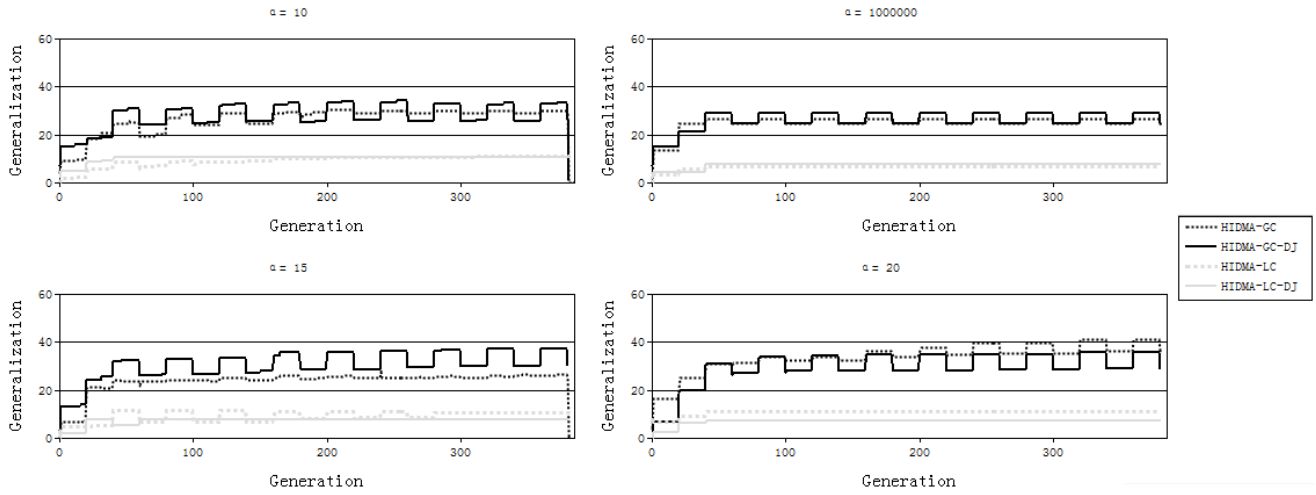
2.AGSsuspect is detected by DCTSmaturation and DCTShighest ,called AMDetect. In this process, new detectors are generated from the highest detectors through affinity maturation. Also, new detectors are randomly generated to implement the TMA.

3 Experiments

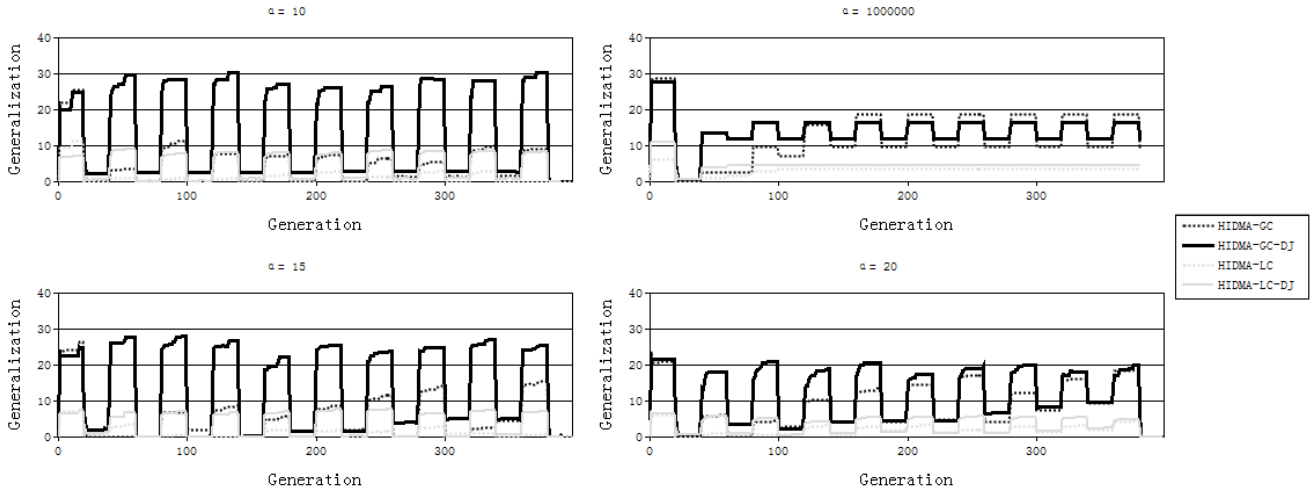
The objective of the experiments is to investigate the generalization capability. Experiments are carried out using the famous benchmark Fisher's Iris Data. Minimal entropy discretization algorithm is used to discretize these data sets [8]. For verifying the adaptive character, nonself data are changed every 20 generations, maxundetectCount=maxg.

In the Iris Data, It has 4 attributes and has total 150 examples with three classes: 'Setosa', ' Versicolour', ' Virginica'. Each class has 50 examples. One of the three types of iris is considered as normal data. The other two are considered anomaly and injected into the algorithm in turn and repeatedly. The proposed algorithm HIDMA-DJ including HIDMA-GC-DJ and HIDMA-LM-DJ, HIDMA-GC and HIDMA-LM runs for 10 times especially with different α which is set to 10,15, 20, 1000000. The max generation maxg=1000000.

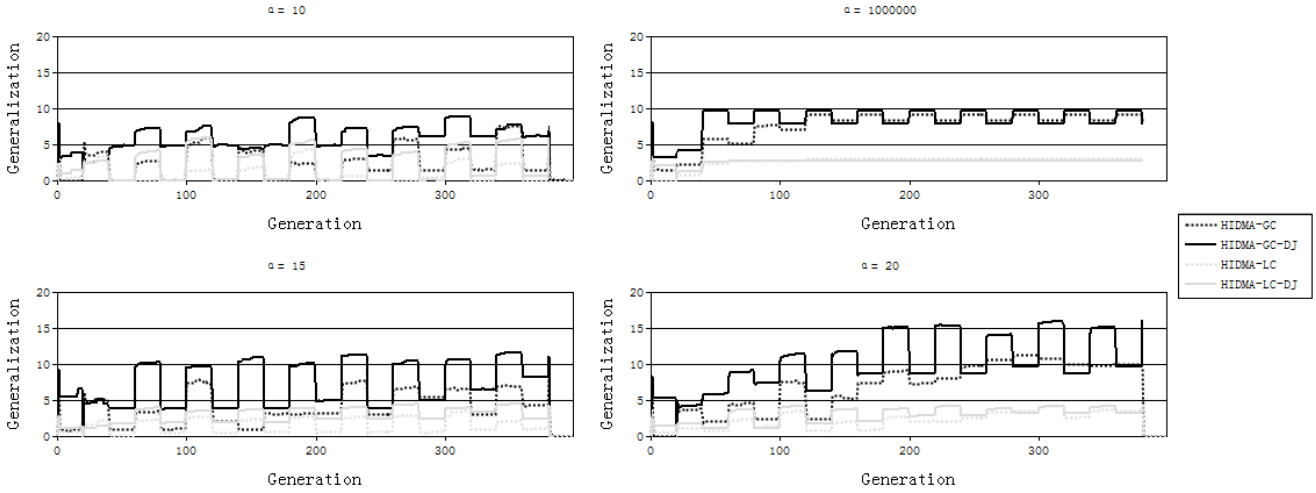
The result of generalization capability is shown in Fig.3. The generalization's value is equal to the quotient when the number of antigens detected by detectors divided by the number of detectors. So the bigger the generalization's value, the more antigens one detector can detect. In Fig.3, the algorithm HIDMA-GC-DJ is more effective in most case.



(a) Setosa as Self Data Set



(b) Versicolour as Self Data Set



(c) Virginica as Self Data Set

Fig. 3. The comparation of algorithm using Iris Data

When the parameter α is 10 or 1000000, it is difficult to distinguish which is more effective between HIDMA-GC and HIDMA-GC-DJ. There are some reasons for it. First, in both algorithms, the detector's lifecycle is not bigger enough to live longer and the detector population is not stable in every generation when α is 10. So it is difficult that the detectors with strong generalization capability are generated. Second, when α is 1000000, every detector can life longer enough and excessive detectors are generated. So the quotient is small which cause the generalization's value for both algorithms is small.

4 Conclusion

In this work, to improve the generalization capability, DeJong's gene expression is used and an augmented HIDMA algorithm (HIDMA-DJ) is proposed. The results show that the generalization capability is improved. But with different parameter α and different data set, the generalization capability is changed a lot, which is required to research on further.

Acknowledgment

This work is supported by National Natural Science Foundation of China 71071145, Zhejiang Provincial Nature Science Foundation Y1110200, Ministry of Science and Technology project 2009GJC20045. Thanks for the assistance received by using KDD Cup 1999 data set [http://kdd.ics.uci.edu/databases / kddcup99/ kddcup99.html]

References:

[1]Hart E., Timmis J.,Application areas of AIS: The past, the present and the future, Journal of Applied Soft Computing,2008,8(1):191-201

[2]Timmis J., etc, An interdisciplinary perspective on artificial immune systems, Evolutionary Intelligence, 2008, 1(1):5-26

[3]Greensmith J., Aickelin U., etc, Information Fusion for Anomaly Detection with the Dendritic Cell Algorithm, Information Fusion, 2010, 11 (1): 21-34

[4]Hofmeyr S. A., An Immunological Model of Distributed Detection and its Application to Computer Security,PhD Dissertation. University of New Mexico, 1999

[5]Jungan Chen,etc.Hybrid Immune Detector Maturation Algorithm with LifeCycle Model. International Conference on Computer Science and its Applications,2009:213-218

[6]Jungan Chen,etc.Improving the Generalization Capability of Hybrid Immune Detector Maturation Algorithm,HAIS12,2012:298-308

[7]Kenneth A. De Jong,etc. Learning Concept Classification Rules Using Genetic Algorithms,IJCAI'91,1991

[8]Dougherty J., Kohavi R. and Sahami M.: Supervised and unsupervised discretization of continuous features. Available http://robotics.stanford.edu/~ronnyk/disc.ps