RESEARCH SOME ALGORITHMS IN MACHINE LEARNING AND ARTIFICIAL IMMUNE SYSTEM, APPLY TO SET UP A VIRUS DETECTION SYSTEM

Vũ Thanh Nguyen[(1)], Nguyen Vinh Kha[(2)], Nguyen Phương Anh[(3)]
University of Information Technology – Vietnam National University, HCM City
6 Quarter, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

Abstract — **The article learns about the components and mechanisms of artificial immune system (AIS). Three key problems of artificial immune system: pattern recognition, negative selection and clonal selection will be presented in essay thoroughly.**

Based on the theoretical results, the authors implemented an adaptive virus detection system. As a smart virus detection system, it is not only capable of detecting known virus patterns but also have the ability to identify new threats. Two popular classifiers, RBF network and SVM, also are integrated to the system to increase the detection ability of the system.

Keywords: **Anti-virus, artificial immune system, artificial intelligent.**

I. INTRODUCTION.

At the moment, there are two main virus detection methods have been applied widely:

Data-based method: Use virus signature to detect harmful files. As polymorphic virus can change their signatures while spreading [1], it is becoming more complicated to extract signatures as well as to detect them.

Behavior-based method: utilize the operating system's application programming interface sequences, system calls or other kinds of behavior characteristics to identify the purpose of a program [2]. Although these approaches have produced promising results, they can produce high rates of false positive errors, an issue which has yet to be resolved [3].

The disadvantages of these methods are the reason of the requirement for some heuristic data-based methods. Among them, AIS is considered as a dynamic and adaptive solution. To increase the capability of AIS, classifiers have been used for the purpose of integration.

Besides the Introduction, the material includes following chapters: II. Overview about artificial immune system; III. Building up a virus detection system; IV. Integrating and V. Evaluation and summary.

II. OVERVIEW ABOUT ARTIFICIAL IMMUNE SYSTEM.

A. The definition of artificial immune system

"Artificial immune system (AIS) can be defined as computational systems inspired by theoretical immunology, observed immune functions, principles and mechanisms in order to solve problems." [4]

AIS has a wide range of application: pattern recognition, fault and abnormal event detection, data analysis, schedule, machine learning…

The general idea of immune algorithms is adjusting the antibody population, leading it to a good solution (convergence of population) as well as ensuring the diversity of population (to avoid cases causing the early convergence). The adjustment is based on the affinity between antibodies with each other and with antigens. The evaluation methods are different for different algorithms.

III. BUILDING UP A VIRUS DETECTION SYSTEM (VDS).

A. Detector

In the VDS, detector has the length of $L = 32$, and is the binary string (bit string)($m = 2$). The sequences

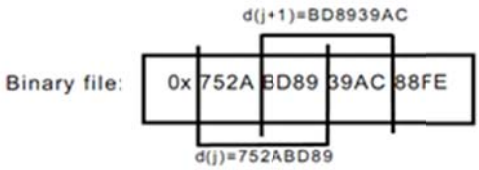

Fig. 1. The representation of data extraction

are extracted from virus files as figure 1.

The binary sequences with the length $L = 32$ will be extracted from the binary file continuously. Two neighbouring fragments have overlap of $L/2$ bits. The direct extraction with high density is for the increase of diversity of detector and for the decrease of missing signs used to identify virus.

B. String matching rule.

The two string matching rule recently used are Hamming and r-Contiguous. In the material, the r-Contiguous will be used.

Both *Hamming* and *r-Contiguous* are able to be adjusted through a threshold r, $0 \le r \le l$; the greater value of *r*, the more stringent condition of matching.

A = 11001010101101010
B = 00101010011010111

Fig. 2. The Hamming matching rule

In the example, the Hamming matching rule is applied to two $L = 16$ fixed length sequences, sequences' characters are from a set of m = 2 distinct elements and the threshold $r = 9$. Two sequences A and B are matching with $r \le 9$.

A = 1100101010111010
B = 0010101001101011

Fig. 3. The r-Contiguous matching rule

C. Negative selection

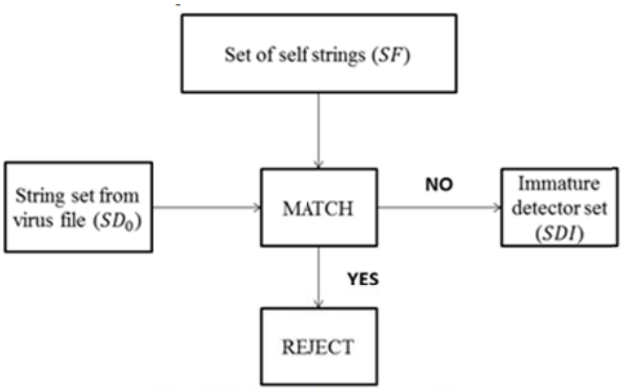

Fig. 4. Negative selection process

Negative selection mechanism is usually used to detect abnormal behaviors or elements.

The mechanism works on the hypothesis that if a detector is capable of recognizing a testing element, which is the self-one, it must be eliminated. This way, the detector survive negative selection are assume to recognize only non-self elements.

This is the process for rejecting abnormal detectors. First, a set of detectors (SD_0) is generated by extracting bit string from virus files. Perform the test for SD_0 by determine the affinity of all detectors with all bit string in self-string set (*SF*), and then eliminate the bit string detector capable of recognizing. If the affinity of a bit string in SD_0 with at least one string of *SF* is greater than or equal to a given affinity threshold *r,* then the string in SD_0 recognizes the self-string and has to be eliminated; otherwise the string belongs to the non-self set and is introduced into the set *SDI*.

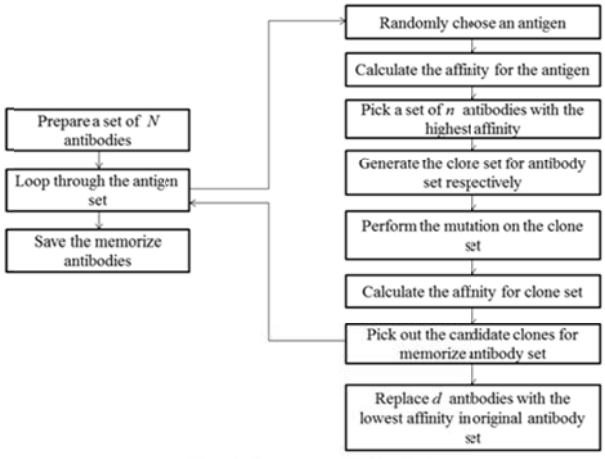

Fig. 5. Process of CLONALG

D. Clonal Selection

Theory of the clonal selection is basically used for improving the AIS in optimizing the calculation and pattern recognition. The used model is executing the simulating progress of antigen-specific affinity maturation of B cells along with the hyper mutation.

The technique of AIS used to simulate the progress of clonal selection and hyper mutation is named CLONALG (CLONal selection ALGorithm). Below are the installation steps of CLONALG

To determine the number of clones for each antibody, we execute arranging the antibodies in ascendant order according to affinity on antigens. Scanning down the list of the arranged antibodies, the number of clones created based on the following formula:

$$numClones = \left[\frac{\beta . N}{i} + 0.5\right] \quad (5)$$

β is clonal factor, N is the size of antibody sets, and i is the index of current antibodies, $i \in [1, n]$. Therefore, the total of clones created from set n is:

$$Nc = \sum_{i=1}^{n}\left[\frac{\beta . N}{i} + 0.5\right] \quad (6)$$

Each clone has a mutation factor α, which is for defining the capability of clone mutation:

$$\alpha = [e^{(-\rho \cdot f)}] \quad (7)$$

ρ is mutation factor, f is the affinity between the original antibodies and the antigens. β is often chosen from (0, 1] and ρ is included in [1, 10].

IV. INTEGRATING.

Using matching comparison for virus detection is the easiest method with reasonable result. However, in long term, it is not a good solution; a more evolutionary one is required.

Classification algorithms can be applied to improve the possibility of detection. For the integration, three important stuffs should be considered:

- Determining the features
- Choosing the classifiers.

- Applying the classifier into the general model of VDS.

In the parts of material, all of the three will be discussed.

A. Max Hamming dangerous level

The best matching value between $x_i^d \in SD$ and $x_j^f \in SF_l$ can be found by using hamming distance. The max Hamming distance can be evaluated by the equation:

$$MHD\left(x_j^f\right) = max\left\{HD(x_i^d, x_j^f)\right\} \quad (7)$$

Where x_j^f is a bit string in the set of bit strings SF_l extracted from the file l, SD is the set of detector bit string.

B. Max r-continuous dangerous level

Max r-continuous bit distance can be obtained by the following equation:

$$MRD\left(x_j^f, r\right) = max\left\{RD(x_i^d, x_j^f, r)\right\} \quad (8)$$

Where $RD(x_i^d, x_j^f, r)$ is the r-continuous bit distance between x_i^d and x_j^f.

C. Determining the features – danger level

We can calculate the danger level of a data fragment x_j^f by the below equation

$$DL\left(x_j^f\right) = \langle MHD\left(x_j^f\right), MRD\left(x_j^f, 12\right), MRD\left(x_j^f, 24\right)\rangle \quad (9)$$

In the equation, $x_i^d \in SD$, DL is the danger level of x_j^f in the set of bit strings SF_l of the file l, $|SF_l| = n^f$. $MHD\left(x_i^d, x_j^f\right)$ is the Max Hamming dangerous level of x_j^f. $MRD\left(x_j^f, 12\right)$ is the Max r-contiguous dangerous level of x_j^f with the threshold 12.

The danger level of file f can be obtained by the following equation:

$$dv_l = \frac{\sum_{j=0}^{n^f} DL(x_j^f)}{n^f} \quad (10) \ [8]$$

dv_l is the affinity vector of file l, typical for the danger level of the file from three different distances. The file's probability of being virus increases with the affinity vector.

D. Classifier algorithms.

At the moment, there are many classifiers able to be integrated with the system. In the scope of the paper, we will consider two popular classifiers: RBF network and SVM.

RBF network classifier

A radial basis function (RBF) network is the special type of neural network using a radial basis function as activation function [5]. Being different from other neural networks, it possesses several distinctive features. A RBF network consists of three layers: input layer, hidden layer and output layer. The input layer broadcasts the coordinates of the input vector to each unit in the hidden layer. Based on the radial basis function, each unit in the hidden layer then produces activation. Finally, each of the units in the output layer computes a linear combination of the activations of the hidden units [6]. RBF network has been applied in many applications including function approximation, data classification, and data clustering.

SVM classifier

SVM are a learning method introduced by Vapnik [7] based on his Statistical Learning Theory and Structural Minimization Principle. Finding the optimal separating hyper plane between the positive and negative examples is the main approach of SVM. The best hyper plane is the one giving the maximum margin between the training examples that are closest to it. Support vectors are the examples that lie closest to the separating hyper plane. After finding the hyper plane, a new example can be classified simply by determining on which side of the hyper plane it is.

E. Applying the classifiers into the general model.

For integrating purpose, AIS and the classifier are installed in one system. The conclusion of AIS is used to initialize the characteristic parameters for the training and testing phases of classifier.

Below is the process of the integration:

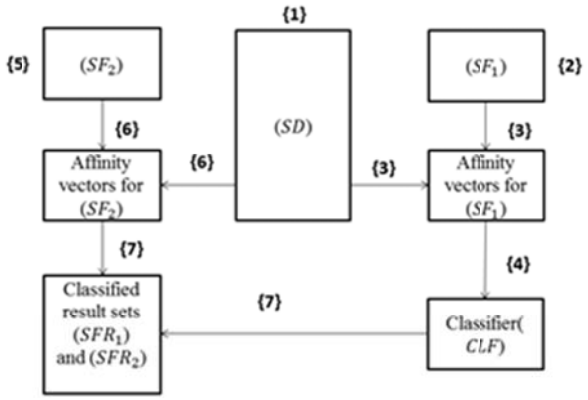

Fig. 6. Process of sequential integration

{1}: Extract data fragments for training process of AIS. Apply the negative and clonal selection for SD_0: The result is a set of mature detectors SD.

{2}: Extract data fragments for training the classifier CLF (rbf network and SVM). Result of the phase is the set SF_1. It contains the bit strings which are labelled as self or non-self.

{3}: Use the detector set SD to calculate the features used in classifier training.

{4}: Train the classifier CLF.

{5}: Extract bit strings for testing by classifier CLF. The result is the set SF_2.

{6}: Calculate the features of elements in SF_2.

{7}: Process to classify SF_2 with CLF. The result contains two sets SFR_1 (virus bit string set) and SFR_2 (benign bit string set)

V. EVALUATION AND SUMMARY.

A. Evaluation

The figure 7 shows the number of files in dataset using in our experiment. Figure 8, 9 and 10 are the average detection rate of not using classifier, using SVM and RBF network respectively.

Training set		Testing set	
Virus Files	Benign Files	Virus Files	Benign Files
869	90	1208	118

Fig 7. The number of files in dataset

Percent of training set used	25% dataset		50% dataset		75% dataset	
Type of files	Virus	Benign	Virus	Benign	Virus	Benign
Training set	78.45 %	76.14%	75.21 %	77.33%	70.64%	78.48%
Testing set	76.98 %	79.65%	78.03 %	77.35%	78.89%	76.29%

Fig 8. The average detection rate when not using the classifier

Percent of training set used	25% dataset		50% dataset		75% dataset	
Type of files	Virus	Benign	Virus	Benign	Virus	Benign
Training set	82.65 %	83.58%	79.54 %	84.66%	76.25 %	84.94%
Testing set	83.33 %	83.04%	84.46 %	82.54%	85.52 %	79.51%

Fig 9. The average detection rate of SVM

Percent of training set used	25% dataset		50% dataset		75% dataset	
Type of files	Virus	Benign	Virus	Benign	Virus	Benign
Training set	85.72%	85.32%	84.54%	86.47%	78.17%	86.96%
Testing set	77.96%	84.06%	83.28%	83.54%	84.27%	81.52%

Fig 10. The average detection rate of RBF network

Fig. 11. The average detection rate of SVM and RBF network in comparison

Through 4 charts in figure 11, the average detection rates when using classifier much higher than not. When using only the matching rule without the classifier, we bypass the advantages of data mining which bring more accuracy to classify the input data through self-learning and self-adapting mechanism. With the matching rule, the detection rates almost depend on the matching threshold so choosing right threshold to bring the best detection rates is complex.

In chart 1, the more files using in data training process, the lower average detection rates of virus file in training set we got in both algorithms. Virus files always contain both virus characteristics and benign codes so that the more files we use in training process, the more useless detectors we put into the detector set. We must increase amount of benign files to reduce amount of useless detector through negative selection process.

In chart 3, the more files using in data training process, the higher average detection rates of virus file in testing set because a large amount of training

files enrich the detector set. But the SVM has more stable detection rates than the RBF when there's a lack of training data.

Fig. 11 shows that RBF network has better performance than SVM. The RBF network algorithm has the advantages of fast learning, high accuracy and strong self-adapting ability with the large amount of training data, so it has the highest detection rate in most datasets. Meanwhile, the SVM algorithm has a stable detection rate and only needs a small quantity of training data to train the classifier.

B. Summary

Based on the knowledge of artificial immune system, the paper proposed the general model for a virus detection system. Using basic theories such as negative selection, clonal selection as well as the r-Contiguous matching rule, a virus detection system has been set up successfully. To increase the competence of detection, two classifiers are integrated to the system.

REFERENCES

[1] Aickelin, U., Greensmith, J., and Twycross, J, "Immune System Approaches to Intrusion Detection—a Review", 2004

[2] Kerchen, P., Lo, R., Crossley, J., Elkinbard, G., and Olsson, R. "Static Analysis Virus Detection Tools for Unix Systems", National Computer Security Conference, 1990.

[3] Hofmeyr, S.Forrest, S.and Somayaji, A. "Intusion Detection Using Sequences of System Calls", Journal of Computer Security, 1998.

[4] L. N. de Castro and J. I. Timmis "Artificial immune systems as a novel soft computing paradigm", Soft Computing 7, 2003.

[5] Broomhead, D and Low, D. "Multivariable functional interpolation and adaptive networks" Complex Systems, 1988.

[6] Yen-Jen Oyang , Shien-Ching Hwang, Yu-Yen Ou, Chien-Yu Chen, and Zhi-Wei Chen, "Data Classification with Radial Basis Function Networks Based on a Novel Kernel Density Estimation Algorithm"

[7] V. Vapnik, "The Nature of Statistical Learning Theory", Springer, 1999.

[8] Rui Chao and Ying Tan, "A Virus Detection System Based on Artificial Immune System", 2009 International Conference on Computational Intelligence and Security.