# Immune Detector Optimization Algorithm with Co-evolution and Monte Carlo

Xi Liang(✉), Jiang Tao, Sun Guanglu, and Zhang Fengbin

School of Computer Science and Technology,
Harbin University of Science and Technology, Harbin 150000, China
xiliang@hrbust.edu.cn

**Abstract.** The detector which is devoted to detect the abnormal events in the immune-based instrusion detection system (IDS) is absolutely necessary. But, some problems in the detector set need to be solved before detection, and at the same time, the research in the security vulnerabilities detector optimization is important. In this paper, inspired by the species' co-evolution in nature and the Monte Carlo method, An algorithm of immune detector optimization is presented: co-evolve among detector subsets, estimate the coverage rate by Monte Carlo to end the optimization. Getting a conclusion by the experimental tests is that the security holes can be fewer by the algorithm, and less detectors can be used to achieve more accurate coverage of non-self-space.

**Keywords:** Instrusion detection system · Artificial immune system
Co-optimization · Detector · Monte Carlo

## 1 Introduction

Intrusion detection system is a significant component of network security. The basic problems in Intrusion detection can be seen two problems: one is that give an element of the network, the other one is that divide it into normal or abnormal data [1]. Being a classical subfield of artificial intelligence, it is a relatively new territory which is the artificial immune system (AIS) that attempts to create some mechanisms in the biological immune system (BIS) which is a self-adaptive, self-organized, and self-learning protection system [2]. The task of IDS can be considered as analogous to the BIS, while both methods are designed for the detection of abnormal behavior which is in violation of the established policy properly. So, many models and methods in AIS are used in the field of intrusion detection. The immune IDS has achieved great successes [3].

The immune detectors are the most important ingredient in immune IDS, which ensures the detection performances, and gets the candidates through Self-setting tolerance training by the NSA primarily [4]. On the based of the representation method of self and detector: binary and real-valued, NSA is devided into binary NSA (BNS) and real-valued NSA (RNS). BNS is hard to handle many application programs which are normal to be expressed in the real-valued space. So that, the present research mainly focuses on the representation of real-valued [5]. However, because of the randomness and incompleteness of candidates, security holes are difficult to solve effectively (the uncovered nonself space), and spending too much time on the detector generation [6].


G. Sun and S. Liu (Eds.): ADHIP 2017, LNICST 219, pp. 334–340, 2018.
https://doi.org/10.1007/978-3-319-73317-3_39

For these problems, using the theory of cooperative evolution of biology and the Monte Carlo method for reference, this paper come up with an immune detector optimization algorithm with co-evolution & Monte Carlo, which uses the subsets of detectors to co-optimization by the representative individuals, and assess the scope of the coverage of detectors by the Monte Carlo method to improve detectors' distribution.

The remaining structure of the article is as follows: Sect. 2 is that we analyze the flaw in the detectors and the results. Section 3 introduces the detector optimization algorithm in detail. The experiment was carried out in Sect. 4. Finally, the Sect. 5 is some concluding remarks by the experiment.

## 2 Problem Analysis

### 2.1 Holes and Overlapping

The detectors have two problems which are a pair of contradictions: holes and overlapping. For a better coverage, the detectors' number should be large enough which can bring about the overlapping. For less overlapping rate, the detectors should be less which can bring about the holes. In the real-valued space, these problems are unavoidable.

### 2.2 Problems of Boundary Detectors

In the boundary between self and nonself region, assignment of each detector's radius is a very difficult question. And the detectors can not cover the boundary well which is too narrow, which is referred to boundary holes problem. A classical solution is enlarging radius of these detectors properly. But the "properly" can not be controlled correctly and lead to the intrusion problems which can increase false alarm rate in detection stage. As it was remarked in a previous column, V-detector with boundary-aware by Zhou solves the intrusion better, but the boundary overlapping is worse.

### 2.3 Multi-area of Self/Detector Set

The self/detector region was almost deemed to be a whole in the real-valued shape-space. However, as a matter of fact, the attribute values of self/detector almost are some statistical data. Therefore, multi areas may make up to be the self/detector region. We should consider this character in optimizing the self/detector for a better result.

## 3 Detector Co-optimization

After analyzing the main problems which are existing detectors, inspired by the co-evolution of species in nature, a detector co-optimization algorithm with co-evolution & Monte Carlo (abbr. DOCEMC) to be raised: the detectors are divided into different subsets, optimize process within every subset taking advantage of the

individuals which are representative in other subsets and select the combination of the every subset to form the final mature detector set in the end. In the process, the Monte Carlo method monitors the coverage of detectors in real time and serves as the "trigger" of algorithm termination. The algorithm can be stated in Fig. 1 and the concrete processes are expatiated as follows:

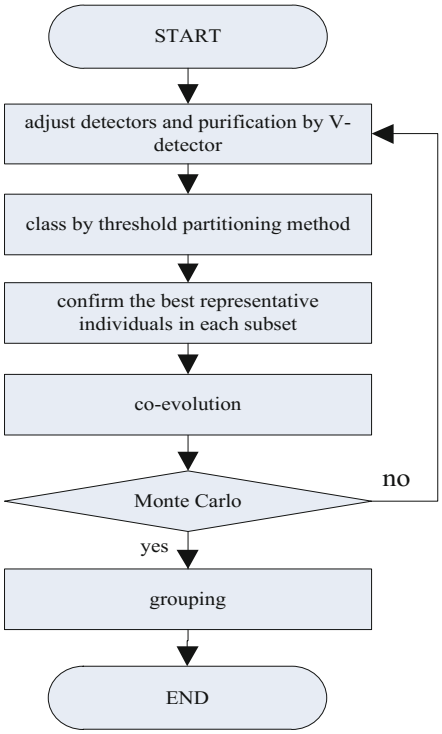


**Fig. 1.** Algorithm flowchart

**Start.** Using random method, a candidate is formed and the initial set of detectors is generated: D by RNS.

**Adjusting Detectors.** For each detector: $d_i$ ($i$ = 1, 2,…, $N_d$), use the closest distance to its self for adjusting its radius:

$$d_i \cdot r = \mathrm{AC}\left(d_i, s_i^{nearest}\right) - s_i^{nearest} \cdot r \tag{1}$$

where AC ( ) is the affinity calculation formula.

**Purification.** Cancel the low-performance samples which are replaced by others using V-detector in every subset.

**Classifying.** Set the original detector to different subsets $D = \{D_1, D_2, \ldots, D_m\}$. The quantity of subsets are $m$ in the D. Divide the detector set by using the threshold partitioning.

**Step 1.** Set a threshold $\Delta$ for distance. The first original partition is $d$ by taking a random detector: $D_1$ $(D = D - \{d\},\ d \in D_1, m = 1,$ the $m$ can be confirmed after classifying).
**Step 2.** For each $D_j$ $(1 \leq j \leq m)$, Check the rest of the detectors: $d_i$ $(d_i \in D - \mathrm{Y}_{j=1}^{m} D_j)$ by RNS. If the $\Delta$ is more than the distance. It means they belong to the same partition $(d_i \in D_j, 1 \leq j \leq m)$. If not, take it as a new partition: $D_{j+1}$, $m+$ $+(d_i \in D_{j+1}, 2 \leq j+1 \leq m)$.
**Step 3.** If $D \neq \emptyset$, go to step 2.

**Choosing Representatives.** Get the central element in the every subset. Afterwards, ensure the individuals which can be best representative to the each of the remaining subsets.

(i) Ensuring the basic element in the every subset. In the every subset, $D_i$, get the average of each attribute and search out the individual which is the nearest to that average vector as the basic element, $d_i$.
(ii) Select the best individuals. In the every subset $D_i$, the best individual which is defined to the each of the remaining subsets is the farthest distance by calculating the distance of the each sample to the each of the remaining subsets: $d_i^j \in D_i$, and $j = 1, 2, \ldots, i-1, i+1, \ldots, m$.

**Co-evolution.** Take advantage of the individuals which can be representative well and the optimization procedure of every subset based on coevolution is realized. In the every subset $D_j$, use the $d_i^j (i = 1, 2, \ldots, j-1, j+1, \ldots, m)$ to count the average vector of $d_j$ and $d_i^j$, becoming candidate $d_0^j$. Deal with RNS for self-established tolerance test. If passing the test, its radius will be ascertained by Formula 19, and examined whether other detectors are covered by affinity calculation: eliminate all those covered; if it is a test failure, delete it.

**Monte Carlo.** If the process is from the formula 2 to the end state, turns to GROUPING.

$$C(D) \approx \frac{\sum_{i=1}^{m} X_D(x_i)}{m - \sum_{i=1}^{m} X_S(x_i)} \tag{2}$$

In the detector set, $X_D(x_i)$ shows the number of points. In the self set, $X_S(x_i)$ shows the number of points:

$$X_D(x_i) = \begin{cases} 1, & \text{if } x_i \in D \\ 0, & \text{if } x_i \notin D \end{cases} \tag{3}$$

$$X_S(x_i) = \begin{cases} 1, & \text{if } x_i \in S \\ 0, & \text{if } x_i \notin S \end{cases} \tag{4}$$

**Grouping.** Put all the subsets $D_j (j = 1, 2, \ldots, m)$ together to be the final set of the mature detectors:

$$D = \mathrm{Y}_{j=1}^{m} D_j \tag{5}$$

## 4 Experiments

This paper detects the availability of the algorithm by two data sets: using the set of 2-dimensional data to test the optimal intuitive performances; making an examination for the detection performances of the final best detector set and initial detector set by Fisher's Iris Data set.

### 4.1 Experiments in Two Dimensional Data Sets

The pentagram data set which is often used and contains 198 samples which in pentacle shape is adopted in this experiment [7]. For all the samples of the experiment, they build the set by themselves, which are shown by Fig. 2(a). After the process of RNS, RNS generates 100 detectors, which are shown by Fig. 2(b). By the figure, we can find many security vulnerabilities. Then, 600 detectors are generated by the same method continuously, which are shown by Fig. 2(c). By the figure, we can find the problem of the security vulnerabilities has been reduced, but more detectors produce more inaccurate points. Finally, we used 100 samples by DOCEMC which mentioned above to optimize the detector set, and the result is shown by Fig. 2(d). By the figure, we can find that the quantity of the detectors are reduced (quantity: 43) obviously and the method solves the security vulnerabilities.

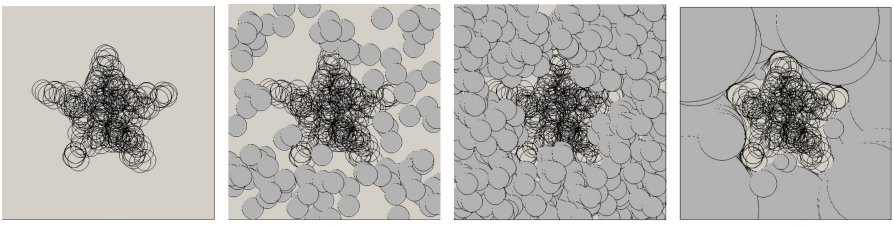


**Fig. 2.** Results of detector distribution: (a) initial self set; (b) detectors by RNS (num: 100); (c) detectors by RNS (num: 600); (d) optimized detectors by DOCEMC (num: 43).

### 4.2 Fisher's Iris Data Set Experiments

The Fisher's Iris data which includes three subsets of data is a famous statistic of Iris flower. Each subset represents one kind of the flower, namely Setosa, Versicolor and Virginica. There are 50 samples in every group, and there are 4 attributes as calyx, calyx width, petal lenth and petal width (units: cm) in each sample. This data set has been used for to the abnormal detection.

Two classes (Versicolor and Virginica) of the data sets are semblable On the distribution through analyzing, however, class Setosa is not the same distribution in spatial. Make the Setosa to be the self-set in the experiment. Veriscolor and Virginica are the exception events. And employ all the data to check the Detector performance. Firstly, produce 100 detectors with RNS as original detector. Secondly, use the algorithm presented in the paper to optimize them. Finally, check the original and optimize detector set with the test sets. Table 1 shows the average of 10 times. As it is shown, RNS-generated intimal detectors have poor performances, while the performences are observably improbed and the detectors have smaller numbers after the optimization.

**Tab. 1.** The comparison between two detector sets in detecting performances

| Algorithm | Detector | Detection rate (%) | | False alarm rate (%) | |
|---|---|---|---|---|---|
| | | Mean | Standard deviation | Mean | Standard deviation |
| RNS | 100 | 58.4 | 7.8 | 8.21 | 5.5 |
| DOCEMC | 37 | 97.9 | 1.4 | 1.97 | 1.9 |

## 5 Conclusions

The optimizing algorithm of the detectors based on Monte Carlo method and co-evolution is proposed in this article. An ideal solution is provided to resolve the deficiencies in real-valued detectors by using the inter-effective relationship between sub-populations to seek the optimal individuals and optimize the subset. The experimental consequences indicate that the algorithm can replace the non-self space with better detecors, sovling the security vulnerabilities and decreasing the quantities of the detecors, making the detector's performance better.

**Acknowledgments.** This article is supported by the Project of Education Department in Heilongjiang Province (12541130). The author also thanks the reviewers for their helpful comments and suggestions which are improved the article.

## References

1. Miao, F., Wang, Z., Guo, Y., et al.: A security threats taxonomy for routing system intrusion detection. In: The 12th International Conference on Computational Intelligence and Security (CIS), pp. 267–270. IEEE (2016)

2. Tabatabaefar, M., Miriestahbanati, M., Grégoire, J.C.: Network intrusion detection through artificial immune system. In: The 11th Annual IEEE International Systems Conference (SysCon), pp. 1–6, April 2017
3. Okamoto, T., Tarao, M.: Toward an artificial immune server against cyber attacks. Artif. Life Robot. **21**(3), 351–356 (2016)
4. Renjie, W., Xiaoling, G., Xiao, Z.: A algorithm of detectors generating based on negative selection algorithm. Lect. Notes Electr. Eng. **375**, 133–139 (2016)
5. Abreu, C.C.E., Duarte, M.A.Q., Villarreal, F.: An immunological approach based on the negative selection algorithm for real noise classification in speech signals. AEU-Int. J. Electron. Commun. **72**, 125–133 (2017)
6. Fouladvand, S., Osareh, A., Shadgar, B., et al.: DENSA: an effective negative selection algorithm with flexible boundaries for self-space and dynamic number of detectors. Eng. Appl. Artif. Intell. **62**, 359–372 (2016). Article in Press
7. UCI Machine Learning Repository: Fisher's Iris Data [DB/OL], 23 December 2009. http://archive.ics.uci.edu/ml/datasets/Iris