An Adaptive Algorithm Based on Adaboost for Mimicry Multimode Decisions

Feng Wang[1], Dingde Jiang[1(✉)], Zhihao Wang[1], and Yingchun Chen[2]

[1] School of Astronautics and Aeronautic, University of Electronic Science and Technology of China, Chengdu 611731, China
jiangdd@uestc.edu.cn
[2] No. 30 Research Institute, CETC, Beijing, China

Abstract. The traditional information security protection cannot prevent the malicious and directed intrusion of the network. To discover potential risks comprehensively, accurately, and timely, the polymorphic heterogeneous executor is constructed to confuse the attacker, called mimicry multimode decision. However, heterogeneous executors are composed of complex hardware, systems and applications, so how to select the optimal combination to face the potential risks becomes a problem. This paper proposes a mimicry multimode decision scheme based on Adaboost machine learning algorithm. The administrator can utilize Adaboost classifier to adaptively select the combination of the most defensible executor, so as to realize mimicry multimode defense and improve the security of applications. Simulation results demonstrate that the adaptive mimicry multimode decision method is promising.

Keywords: Information security · Mimicry multimode decision · Heterogeneous executor · Adaboost

1 Introduction

While the improvement of information technology and the increasing coverage of information system [1–3], the information disclosure is no longer only the loss of economic interests to the society, enterprises and individuals [4]. The importance of information security has gradually risen. To effectively cope with the increasingly complex information threats and attacks, Technologies like Intrusion Detection system (IDS), vulnerability scanning system, firewall, are widely used [5]. However, these traditional security protection systems still have insufficiencies. IDS can only detect the attack, with high non-response rates and the rate of false positives. The existing firewall technology cannot solve the malicious intrusion behavior of the internal network [6]. In addition, these defense techniques are passive, often focusing on localized threat information. At the same time, considering the polymorphic and heterogeneous service system, these self-directed working mechanisms are unable to find potential risks comprehensively, accurately and timely [7]. As a result, the detection results of existing technologies are

H. Song and D. Jiang (Eds.): SIMUtools 2020, LNICST 369, pp. 127–136, 2021.
https://doi.org/10.1007/978-3-030-72792-5_12

greatly deviated, which is not conducive to helping administrators to formulate security defense strategies. At present, the architecture of information system is polymorphic and heterogeneous, which not only includes the heterogeneity of hardware such as CPU, but also exists the heterogeneity of operating system [8, 9]. Therefore, the storage, transfer and elimination of all kinds of services on heterogeneous platforms may be targeted at any time. Backdoor, virus, and Trojans are emerging in endlessly [10]. The traditional information system defense methods can no longer meet the needs of existing information system [11–13]. The lack of security perception ability now is the biggest threat to the security protection of key services.

Aiming at the above problems, many researchers are studying the mimicry defense system. The current information network is a huge and time-varying system, different kinds of network behaviors have different hidden dangers [14–17]. Aiming at the protection of execution applications, the heterogeneous executors are constructed to confuse the attacker. This kind of protection is called mimicry multimode decision. It will turn the vulnerability backdoor of the specific applications into the uncertain attack effect at the system level. As shown in Fig. 1, Different platforms are combined into a heterogeneous executor set. The administrator selects the currently most secure executor to load the application and extend it to all executors to confuse the attacker. This mechanism can make the target system from the attacker's perspective and present an "uncertainty" effect, so as to effectively improve the level of protection system. Among them, how to efficiently select heterogeneous executor and the adaptive transformation is the key to realize intelligent multimode decision. There have been several research about mimicry multimode decision. Zheng et al. [18] analyzed two background mimicry defense method: the semantic modeling method and the semantic consistency analysis method. They proposed a novel software defined networking (SDN) mimic server defense method, which could effectively modify the active defense efficiency of malicious attacks on Web servers. Authors in [19] proposed a random seed scheduling method considering historical confidence for maximum heterogeneity and QoS. The proposed method could achieve a good dynamic balance between security, dynamics, and QoS. Although some mimicry defense methods were proposed in the above studies, they were complex and lack of adaptability. These methods were difficult to be effective in environments where the best heterogeneous implementations need to be updated at all times.

Machine learning algorithm can keep the continuous learning of decision and improve the accuracy. Consider the above defects, we propose a mimicry multimode decision scheme based on Adaboost machine learning algorithm in this paper. Under the management of the controller, the heterogeneous actuator can use Adaboost model to adaptively select the combination of the currently most defensible executors to face the potential attack threat, so as to realize mimicry multimode defense and improve the security of the important applications.

The following is organized as follows. Section 2 constructs the mathematical model of Adaboost algorithm and then proposes the mimicry multimode decision scheme. The simulation results and analysis are expressed in Sect. 3. We then conclude our work in Sect. 4.

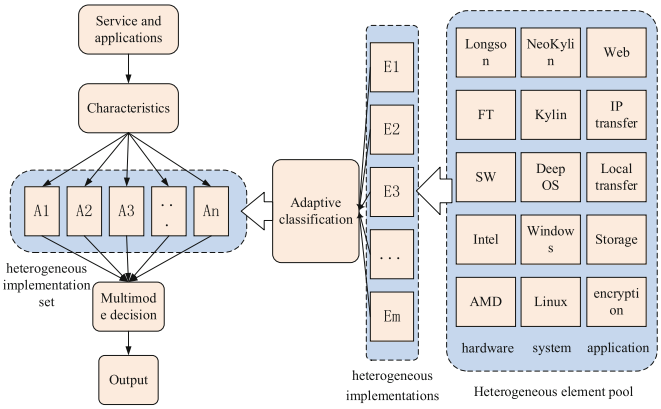

Fig. 1. The mimicry multimode decision architecture.

2 Adaboost Model

In this section, we mainly consider how to carry out mimicry multimode decision using Adaboost classifiers. First, we analyze the classification indicators needed. Subsequently, we established the Adaboost classification model.

As shown in Fig. 1, the heterogeneous element pool includes heterogeneous hardware such as Longson, Intel, AMD, etc. It also includes heterogeneous systems such as Kirin system, Deep OS system, Windows system, Linux system and so on. It also includes heterogeneous applications such as Web services, IP transport services, Local transport services, Storage services, Encryption services and so on. The combination of different platforms has different capabilities of risk resistance. We can evaluate the capabilities of different combinations and set a label based on specific requirements. For example, we can divide all the combinations into four levels: Safety performance 1, 2, 3, and 4. The higher the value, the higher the resistance capabilities. So the problem is how to quickly and efficiently select the most appropriate combination to implement mimicry multimodal decision. The traditional approach, which considers only a limited number of combinations, is weak in the face of endless risks. Therefore, we need to use machine learning algorithms, like Adaboost model, to help administrators adaptively choose the best combination.

AdaBoost is an adaptive classification algorithm, as Fig. 2 shows. A fleet of weak classifiers is trained then combined to form a strong classifier to complete the training framework. The adaptability is that the weight of the data from the samples wrongly classified by the former weak classifier will increase. The weight of properly classified samples will be reduced. These are utilized to build the next weak classifier. The final classifier will only be determined in the following cases: if the error coefficient is small enough or reaches the maximum number of iterations.

Firstly, we obtain a collection of $N + P$ heterogeneous combinations. Then we get the result $R_1, R_2, \cdots, R_N, \cdots, R_{N+P}$. Each R_i has n items that can be represented by $R_i = [v_1, v_2, \cdots, v_n]^T$, where v_i is the code name of the platform. Each result R_i is then

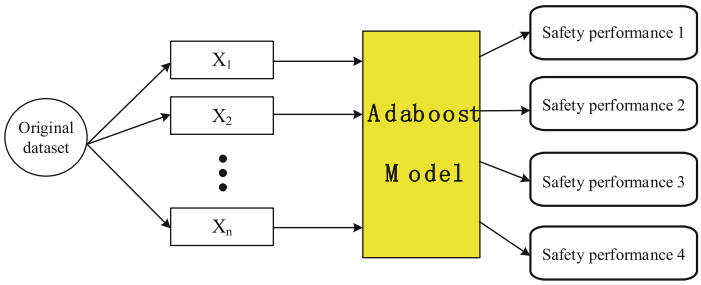

Fig. 2. The Adaboost model.

pre-processed to get the result G_i and its safety performance label m. Finally, we get the dataset D:

$$D = \{(G_1, m_1), (G_2, m_2), \cdots, (G_{N+P}, m_{N+P})\} \tag{1}$$

where $G_i \in \chi \subseteq N^n$, $m_i \in \{1, 2, 3, 4\}$. We divide the former N data into the training dataset D_N. We set the other P data as the testing dataset D_P. Now the training dataset D_N is utilized for model training. We divide D_N into two datasets, where class I is $m = 1, m = 2$ and class II is $m = 3, m = 4$. Then dataset $D_N^{(1)}$ with current classification label is shown as:

$$D_N^{(1)} = \{(G_1, m_1^{(1)}), (G_2, m_2^{(1)}), \cdots, (G_N, m_N^{(1)})\} \tag{2}$$

where the classification features are:

$$m_i^{(1)} = \begin{cases} 1, \ m \in \{1, \ 2\}; \\ 0, \ m \in \{3, 4\}; \end{cases} \tag{3}$$

Then, we configure the weight value of each G_i in $D_N^{(1)}$. The initial weight value of each training item is uniform. We can calculate that the initial weight value $W_1(i)$ of the training data set $D_N^{(1)}$ is:

$$W_1(i) = (w_1, w_2, \cdots w_N), \ w_i = 1/N \tag{4}$$

Then, the fleet of weak classifiers begin to be trained in t times iterative. The training steps are shown as follows:

Step 1: Initialing the weight value W_t of $D_N^{(1)}$. Then, design the t weak classifier:

$$H_t(x) : G \to \{1, 0\} \tag{5}$$

Step 2: Using the current weak classifier to divide the training dataset $D_N^{(1)}$. The classification error is calculated:

$$e_t = P(H_t(G_i) \neq m_i) = \sum_{i=1}^{N} w_{ti} I(H_t(G_i) \neq m_i) \tag{6}$$

Step 3: The weight value v of current weak classifier in the final strong classifier is calculated as follows:

$$v_t = \frac{1}{2}\ln(\frac{1-e_t}{e_t}) \tag{7}$$

Then the t weak classifier is obtained:

$$f_t(x) = v_t H_t(x) \tag{8}$$

Step 4: Update the weight value of the training dataset $D_N^{(1)}$:

$$W_{t+1} = \frac{W_t(i)\exp(-v_t m_i H_t(x_i))}{Z_t} \tag{9}$$

where Z_t is the normalization constant:

$$Z_t = 2\sqrt{e_t(1-e_t)} \tag{10}$$

Assemble existing weak classifiers:

$$f(x) = \sum_{t=1}^{T} v_t H_t(x) \tag{11}$$

Then we evaluate the current strong classifier. If the classification error is below the threshold, the iteration is over, and the final strong classifier is obtained as:

$$H_{strong} = sign(f(x)) = sign\left(\sum_{t=1}^{T} v_t H_t(x)\right) \tag{12}$$

Otherwise, enter the $t+1$ iteration until the classification error reaches the threshold.

After obtaining the first strong classifier, we make all the data $D_N^{(1)}$ pass through the strong classifier, and pick out the data with $H_{strong}(X_i) > 0$, which represent the data classified into safety performance 1 and safety performance 2. The remaining data is safety performance 3 and safety performance 4. Analogously, the classification feature for safety performance 1 and safety performance 3 is labeled as 1, the classification feature for safety performance 2 and safety performance 4 is labeled as 0. Repeat the training process once again and the 2nd and 3rd layer strong classifiers will be obtained:

$$\begin{aligned} H_{strong}^{(2)} &= sign(f^{(2)}(x)) = sign\left(\sum_{t=1}^{T} v_t H_t^{(2)}(x)\right) \\ H_{strong}^{(3)} &= sign(f^{(3)}(x)) = sign\left(\sum_{t=1}^{T} v_t H_t^{(3)}(x)\right) \end{aligned} \tag{13}$$

Then, the first time of Adaboost classifier training is complete and the classifier will be tested later. According to the above, process of classifying the safety performance is shown in Fig. 3. The detailed steps are expressed in Algorithm 1.

Algorithm 1 Adaboost based mimicry multimode decision

Input: Table O of original items.

Input: Adaboost classifier $H_1(x), H_2(x), H_3(x)$.

Output: Safety performance vector: SP[N]

for j = 1 to n **do**
  SP [j] = 0 ;
  if sign(H_1 (O[j] > 0) **then**
    if sign(H_2 (O[j] > 0) **then**
      SP[j] = 1;
    else
      SP[j] = 2;
  else if sign(H_1 (O[j]< 0) **then**
    if sign(H_3 (O[j] > 0) **then**
      SP[j] = 3;
    else
      SP[j] = 4;
end for
return SP[N]

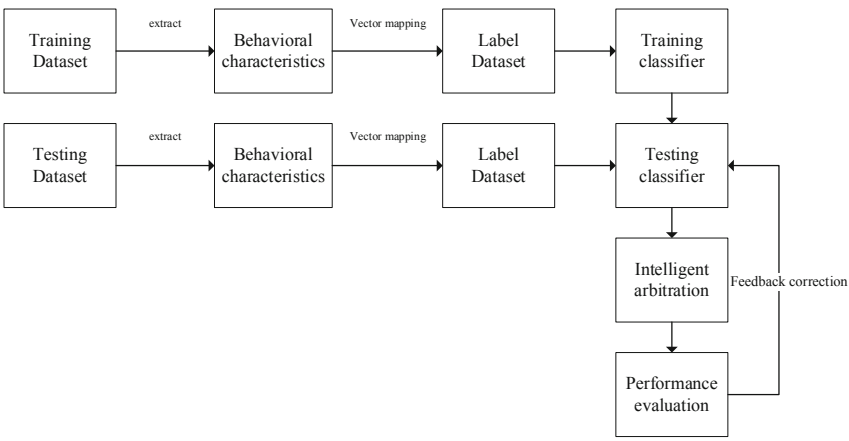

Fig. 3. The Adaboost-based mimicry multimode decision architecture.

3 Simulation Results

Next, we verify the proposed Adaboost classification model for mimicry multimode decision. We mark different types of hardware, systems and applications to values 1–6. Meantime, we make a high defense level environment configuration table and added security performance 1–4 labels for different combinations according to this table. All the preprocessed items are divided into the training dataset and the test dataset of the Adaboost model, and the label are considered as the output result of the model and the reference of the security performance. We first evaluated the model training accuracy under different training samples. Then, we compared the model training complexity of Adaboost algorithm and other common machine learning algorithms. We implement all the algorithms in Matlab 2017b and on the Window10 platform of 3.6 GHz CPU and 32.00 GB RAM. The simulation results are as follows.

We first examined the training results of adaboost model under 100 items. We randomly divided 70 items into a training set, 30 items into a test set, and set the number of weak classifiers to 20. Figure 4 shows the model training results, taking the average value after several experiments. As can be seen from Fig. 4, with the increase of the number of weak classifiers, the training error decreases gradually, and it drops to 0 when the number of weak classifiers reaches 20. At the same time, the test error decreases with the increase of weak classifier, and the final test error is 6%. This shows that Adaboost model can better identify the security performance of different platforms and help realize mimicry multimode decision.

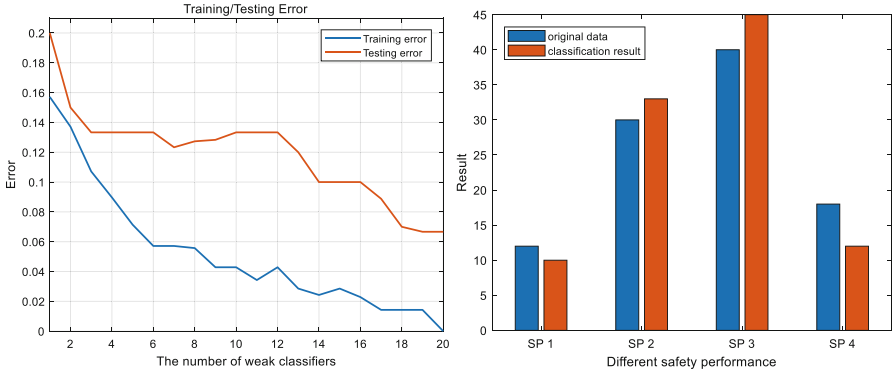

Fig. 4. Classification results and training/testing errors of Adaboost model with 100 items.

Then, we made these 100 items as input to detect the performance of mimicry multimode decision, and the results are shown in the right of Fig. 4. It can be seen that the model can well distinguish the platform combinations with different security performance, and the classification error is 10%. These results show that the training model performs well under 100 items.

In order to verify the adaptability of the proposed model, we randomly selected 200 test items to verify the model classification accuracy at different scales. We randomly divided 150 items into a training set, 50 items into a test set, and set the number of weak

classifiers to 20. Figure 5 shows the model training results, taking the average value after several experiments. As can be seen from Fig. 5, with the increase of the number of weak classifiers, the training error decreases gradually, and it drops to 3% when the number of weak classifiers reaches 20. At the same time, the test error decreases with the increase of weak classifier, and the final test error is 4%. These results show that the training accuracy of the model increases with the increase of training samples. Although the error is larger when the number of weak classifiers is less, the model training error decreases rapidly with the increase of the number of weak classifiers. It shows that the model has a better classification performance under 200 items. Then, we made these 200 items as input to detect the performance of mimicry multimode decision, and the results are shown in the right of Fig. 5. It can be seen that the model can well distinguish the platform combinations with different security performance, and the classification error is 7%. These results show that the proposed mimicry multi-mode decision model has good learning-ability and self-adaptability under the complex and changeable environment.

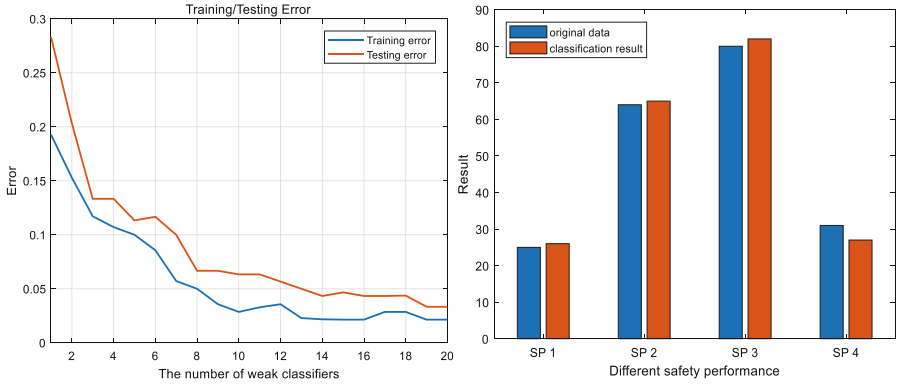

Fig. 5. Classification results and training/testing errors of Adaboost model with 200 items.

In Fig. 6, the time complexity of different machine learning algorithms, like BPNN, SVM and Adaboost, are compared. We utilize the same 200 items to train the two-hidden-layer BPNN and SVM model. For fair comparison, the number of support vectors of the SVM is 20. Similarly, the node number in each BPNN hidden layer is 20. The number of Adaboost weak classifiers is set as 20 for fair comparison. We can find out that as the number of training items increasing, the training time of both BPNN and SVM algorithm are higher than that of Adaboost. The time complexity of upper bound SVM method is a few magnitude orders higher than other three methods. Considering the mimicry multimode decision model needs to keep learning and adaptability in a changing environment, the lightweight Adaboost model that has better performance on time complexity is the optimal machine learning algorithm for mimicry multimode decision.

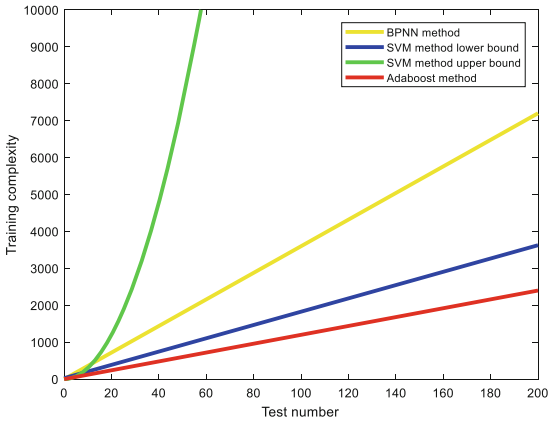

Fig. 6. The comparison of training complexity.

4 Conclusion

The performance of mimicry multimode decision directly affects the intrusion tolerance ability and operation efficiency of the executive system. This paper proposes a mimicry multimode decision scheme based on Adaboost machine learning algorithm. The heterogeneous actuator pool can adaptively select the combination of the most defensible executor using Adaboost classifiers to realize mimicry multimode defense and improve the security of the applications. Simulation results show that the proposed lightweight method can keep good self-learning and adaptability.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (No. 61571104), the Sichuan Science and Technology Program (No. 2018JY0539), the Key projects of the Sichuan Provincial Education Department (No. 18ZA0219), the Fundamental Research Funds for the Central Universities (No. ZYGX2017KYQD170), the CERNET Innovation Project (No. NGII20190111), the Fund Project (Nos. 61403110405, 315075802), and the Innovation Funding (No. 2018510007000134). The authors wish to thank the reviewers for their helpful comments.

References

1. Jiang, D., Wang, Y., Lv, Z., Wang, W., Wang, H.: An energy-efficient networking approach in cloud services for IIoT networks. IEEE J. Sel. Areas Commun. **38**(5), 928–941 (2020)
2. Jiang, D., Wang, Y., Lv, Z., Qi, S., Singh, S.: Big data analysis based network behavior insight of cellular networks for industry 4.0 applications. IEEE Trans. Ind. Inform. **16**(2), 1310–1320 (2020)
3. Qi, S., Jiang, D., Huo, L.: A prediction approach to end-to-end traffic in space information networks. Mob. Netw. Appl. (2020)
4. Rassouli, B., Rosas, F.E., Gündüz, D.: Data disclosure under perfect sample privacy. IEEE Trans. Inf. Forensics Secur. **15**, 2012–2025 (2020)

5. Han, X., Huang, H., Wang, L.: F-PAD: private attribute disclosure risk estimation in online social networks. IEEE Trans. Dependable Secure Comput. **16**(6), 1054–1069 (2019)
6. Husseis, A., Liu-Jimenez, J., Goicoechea-Telleria, I., et al.: A survey in presentation attack and presentation attack detection. In: Proceedings of ICCST 2019, pp. 1–13 (2019)
7. Huo, L., Shao, P., Ying, F., et al.: The research on task scheduling algorithm for the cloud management platform of mimic common operating environment. In: Proceedings of DCABES 2019, pp. 167–171 (2019)
8. Wang, F., Jiang, D., Qi, S.: An adaptive routing algorithm for integrated information networks. China Commun. **7**(1), 196–207 (2019)
9. Huo, L., Jiang, D., Qi, S., et al.: An AI-based adaptive cognitive modeling and measurement method of network traffic for EIS. Mob. Netw. Appl. (2019)
10. Liang, H., Chen, F., Ni, S., et al.: Cloud security in space communication network. In: Proceedings of ICCC 2019, pp. 1053–1057 (2019)
11. Jiang, D., Wang, W., Shi, L., Song, H.: A compressive sensing-based approach to end-to-end network traffic reconstruction. IEEE Trans. Netw. Sci. Eng. **7**(1), 507–519 (2020)
12. Jiang, D., Huo, L., Lv, Z., Song, H., Qin, W.: A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. IEEE Trans. Intell. Transp. Syst. **19**(10), 3305–3319 (2018)
13. Wang, Y., Jiang, D., Huo, L., Zhao, Y.: A new traffic prediction algorithm to software defined networking. Mob. Netw. Appl. (2020)
14. Jiang, D., Huo, L., Song, H.: Rethinking behaviors and activities of base stations in mobile cellular networks based on big data analysis. IEEE Trans. Netw. Sci. Eng. **7**(1), 80–90 (2020)
15. Jiang, D., Huo, L., Li, Y.: Fine-granularity inference and estimations to network traffic for SDN. PLoS ONE **13**(5), 1–23 (2018)
16. Jiang, D., Zhang, P., Lv, Z., et al.: Energy-efficient multi-constraint routing algorithm with load balancing for smart city applications. IEEE Internet Things J. **3**(6), 1437–1447 (2016)
17. Jiang, D., Li, W., Lv, H.: An energy-efficient cooperative multicast routing in multi-hop wireless networks for smart medical applications. Neurocomputing **220**(2017), 160–169 (2017)
18. Zheng, J., Wu, G., Wen, B., et al.: Research on SDN-based mimic server defense technology. In: Proceedings of ICAICS 2019, pp. 163–169 (2019)
19. Wu, Z., Wei, J.: Heterogeneous executors scheduling algorithm for mimic defense systems. In: Proceedings of CCET 2019, pp. 279–284 (2019)