Hindawi
Mobile Information Systems
Volume 2020, Article ID 8844116, 9 pages
https://doi.org/10.1155/2020/8844116

Research Article

An Intrusion Detection Scheme Based on Repeated Game in Smart Home

Rui Zhang,[1] Hui Xia,[2] Shu-shu Shao,[1] Hang Ren,[1] Shuai Xu,[1] and Xiang-guo Cheng[1]

[1]*The College of Computer Science and Technology, Qingdao University, Qingdao 266100, China*
[2]*The College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China*

Correspondence should be addressed to Hui Xia; xiahui@qdu.edu.cn and Xiang-guo Cheng; 15964252399@163.com

Received 25 June 2020; Revised 17 August 2020; Accepted 28 August 2020; Published 17 September 2020

Academic Editor: Vinod Karar

Smart Home brings a new people-oriented home life experience. However, the edge devices in this system are facing severe threats such as data security and equipment safety. To solve the above problems, this paper proposes an intrusion detection scheme based on repeated game. We first use the K-Nearest Neighbors (KNN) algorithm to classify edge devices and equip the intrusion detection system to cluster heads. Secondly, we use the regret minimization algorithm to determine the mixed strategy Nash equilibrium of the one-order game and then take a severe punishment strategy to domesticate malicious attackers. Thirdly, the intrusion detection system can detect malicious attackers by reduction of payoff. Finally, the detailed experimental results show that the proposed scheme can reduce the loss of attacked intrusion detection system and then achieve the purpose of defending against the attacker.

1. Introduction

Internet of things (IoT) is entering people's lives and makes the production and life of human beings more intelligent and convenient. Smart Home is a typical application of the IoT [1]. Smart Home integrates integrated wiring technology and network communication technology and is an effective management system [2]. However, Smart Home is facing severe security threats such as data security and device security [3]. The distribution of edge devices is too scattered to apply security technologies in a Smart Home. Besides, some equipment uses outdated versions that are unable to remotely upgrade weaknesses and vulnerabilities, making Smart Home devices vulnerable to attacks. For instance, equipment such as cameras and smart thermostats collect information about people's daily lives which can be traced directly or indirectly back to the person. Once the data of Smart Home devices is stolen, users' private information will be disclosed. Therefore, it is urgent to design an effective security protection scheme to ensure user data security in the Smart Home.

Intrusion detection technology is a method to resist the attacker invasion, which can monitor, analyze, and deal with a variety of intrusions without affecting network performance as much as possible to improve the ability of networks to deal with external threats. According to the technology used, intrusion detection technology can be divided into three categories: anomaly detection, misuse intrusion detection, and hybrid intrusion detection. The abnormal detection technology can detect the new intrusion, but it is difficult to establish the attacker's behavior model [4]. Misuse detection technology has high detection accuracy, but it is difficult to collect and update intrusion information [5]. Hybrid intrusion detection technology combines misuse detection and anomaly detection, inherits the advantages of both, improves the detection rate, and decreases false positive rate [6]. To sum up, the existing intrusion detection technologies mainly have the following shortcomings: the volume of data is too difficult to process and the data dimension is too high to be reduced.

Inspired by the above schemes, this paper models interactions between attackers and intrusion detection systems as the repeated game and proposes an intrusion detection scheme based on repeated game to protect the security of Smart Home. The main contributions are as follows:

(1) To reduce the cost of equipping the intrusion detection system, this paper uses the K-Nearest Neighbors (KNN) algorithm to classify edge devices and equips the intrusion detection system for cluster heads to achieve the purpose of protecting Smart Home system.

(2) To defend against attackers, we build interactions between attackers and intrusion detection systems as a repeated game model, use the regret minimization algorithm to determine the mixed strategy Nash equilibrium of this game, and set the severe punishment mechanism to force the attacker to take good action.

(3) For the part of the simulation experiment, we compare the proposed scheme with Winner, ALL-S, ALL-P, and ALL-R with three factors: the intrusion detection rate, the attacker's payoff, and the intrusion detection system's payoff. The experimental results show that the proposed scheme can resist attackers.

The remainder of this paper is organized as follows: Section 2 describes the representative achievements of intrusion detection technology. We propose an intrusion detection scheme based on repeated game in Smart Home in Section 3. Section 4 shows the performance of intrusion detection scheme based on repeated game. Finally, Section 5 summarizes the possible expansion and research directions in the future.

2. Related Work

Intrusion detection technology [7] can be divided into three types: anomaly detection, misuse detection, and hybrid intrusion detection. This section mainly summarizes two kinds of techniques of anomaly detection and misuse detection.

The anomaly intrusion detection [8] takes the intrusion activity as a subset of the anomaly activity, which is divided into feature selection-based anomaly detection, Bayesian inference-based anomaly detection, and pattern prediction-based anomaly detection. The feature selection-based anomaly detection is to accurately predict or classify detected intrusions by selecting a subset of metrics that can detect intrusions [9, 10]. However, the metric set cannot encompass all the various intrusion types; and the pre-identified specific metric set may miss intrusions in a particular environment alone. The Bayesian inference-based anomaly detection is to judge whether the system has an intrusion event by measuring the variable [11, 12]. However, this method requires correlation analysis of each variable for determining the relationship between each variable and the intrusion event. The pattern prediction-based anomaly detection considers the sequence of intrusion events and their correlation [13, 14], but the unrecognized behavior pattern is judged as an abnormal event in this method.

Misuse intrusion detection [15, 16] detects intrusion events by matching the defined intrusion pattern with the observed intrusion behavior, which can be divided into contingent probability-based misuse intrusion detection, state transition analysis-based misuse intrusion detection, and keyboard monitoring-based misuse intrusion detection. The contingent probability-based misuse intrusion detection maps the intrusion to an event sequence and then infers the intrusion occurrence by observing the event [17, 18]. However, in this method, the prior probability is hard to give, and the event independences are hard to be satisfied. The state transition analysis-based misuse intrusion detection regards an attack as a series of state transitions of monitored systems [19, 20]. However, the attack mode can only describe the sequence of events and is not suitable for describing complicated events. The keyboard monitoring-based misuse intrusion detection assumes that the intrusion corresponds to a specific keystroke sequence pattern and then monitors the user keystroke pattern and matches this pattern with the intrusion pattern to detect intrusion [21, 22]. But this approach, without operating system support, lacks a reliable way to capture users' keystrokes, and users can easily cheat the technique by using alias commands.

To solve the above problems, we no longer detect the intrusion based on the characteristics of the attacker but consider intrusion detection system's payoff; that is, the intrusion detection system detects the attacker invasion by observing its payoff decrease.

3. Intrusion Detection Scheme Based on Repeated Game

This section describes how the intrusion detection system detects the attacker's malicious action and how to educate the malicious attackers to take good strategy. The notations definitions are shown in Table 1.

3.1. One-Order Game

3.1. One-Order Game. In Smart Home, due to a large number of edge devices and limited service capacity [23, 24], it is impossible to run the intrusion detection system on each edge device, so we need to design a strategy to allocate the intrusion detection system on the edge device. We first use the clustering algorithm to divide edge devices into multiple clusters and then configure intrusion detection system for each cluster-head node in Smart Home [25, 26]. Each cluster has a cluster-head node and several member nodes. The former is mainly responsible for information forwarding and executing the intrusion detection program within the cluster, and the latter is responsible for collecting information and passing the information to the cluster-head node [27, 28]. Suppose that there are N edge devices, which are divided into k clusters by KNN algorithm, $C_1, C_2, \ldots, C_k$. We assume that an attacker can attack one cluster head at a time and model interactions between the intrusion detection systems and attackers as a one-order game model. That is,

Table 1: Notations definitions.

Notations	Definition
C_i	The ith cluster head
S	Attackers and intrusion detection systems' strategy space
c_i	The cost of attacking cluster heads C_i
c_i'	The cost of attacking cluster heads C_i after T times
r_i	The cost of persistently protecting cluster heads C_i
r_i'	The cost of protecting cluster heads C_i after T times
p_a^i	The payoff of attacking cluster heads C_i
p^{di}	The payoff of intrusion detection systems against attacks
M	The strategy matrices of attacker and intrusion detection system
X	Attackers' payoff matrix
Y	Intrusion detection systems' payoff matrix
U_e	The cumulative payoff of player e
δ	The discount factor which measures how much players value future payoffs

$$G_{\text{one-order}} = (P, S, U), \tag{1}$$

where P is the player in one-order game, that is, the intrusion detection system and the attacker, $P = (a, d)$. S is the strategy space, $S = (A_a, D_d)$, and U is the player's payoff. The attacker has four strategies, $A_a = (a_1, a_2, a_3, a_4)$. a_1 refers to the fact that attackers do not attack any cluster heads; a_2 refers to the fact that attackers attack the cluster-head node C_i; a_3 refers to the fact that attackers attack cluster heads C_i after T times; a_4 refers to the fact that attackers attack the cluster-head node C_j. Also, the intrusion detection system has four strategies, $D_d = (d_1, d_2, d_3, d_4)$. d_1 refers to the fact that intrusion detection systems do not protect any cluster heads; d_2 refers to the fact that intrusion detection systems protect the cluster head C_i; d_3 refers to the fact that intrusion detection systems protect cluster heads C_i after T times; d_4 refers to the fact that intrusion detection systems protect the cluster head C_j. Therefore, the strategy profile of attacker and intrusion detection system can be defined as

$$M = \begin{bmatrix} (a_1, d_1) & (a_1, d_2) & (a_1, d_3) & (a_1, d_4) \\ (a_2, d_1) & (a_2, d_2) & (a_2, d_3) & (a_2, d_4) \\ (a_3, d_1) & (a_3, d_2) & (a_3, d_3) & (a_3, d_4) \\ (a_4, d_1) & (a_4, d_2) & (a_4, d_3) & (a_4, d_4) \end{bmatrix}. \tag{2}$$

The row represents the attacker's strategy and the column represents the intrusion detection system's strategy in M. Suppose that U_a and U_d are the payoffs of attackers and intrusion detection systems, respectively. Thus,

$$G = (a, d, A_a, D_d, U_a, U_d), \tag{3}$$

where a refers to the attacker and d refers to the intrusion detection system. The strategy profile $M_{22} = (a_2, d_2)$ refers to the fact that the attacker does not attack the cluster head, whereas the intrusion detection system protects the cluster head. At this time, the attacker gains the payoff 0 at the cost of c_i, $U_a = -c_i$, and the intrusion detection system at the cost of r_i to gain the payoff p_i, $U_d = p_i - r_i$. Similarly, we can get the payoff matrix of attackers and intrusion detection systems, as shown in X and Y:

$$\begin{gathered} X = \begin{bmatrix} 0 & 0 & 0 & 0 \\ p_a^i - c_i & -c_i & p_a^{i'} - c_i & p_a^i - c_i \\ p_a^{i'} - c_i' & -c_i' & p_a^{i'} - c_i' & p_a^{i'} - c_i' \\ p_a^j - c_j & p_a^j - c_j & p_a^j - c_j & -c_j \end{bmatrix}, \\ Y = \begin{bmatrix} 0 & -r_i & -r_i' & -r_j \\ -p_a^i & p_a^i - r_i & p_d^{i'} - r_i' & -r_j \\ -p_a^{i'} & p_d^{i'} - r_i & p_d^{i'} - r_i' & -r_j \\ -p_a^j & -r_i & -r_i' & p_d^j - r_j \end{bmatrix}, \end{gathered} \tag{4}$$

where c_i is the cost of attacking cluster heads C_i, c_i' is the cost of attacking cluster heads C_i after T times, r_i is the cost of persistently protecting cluster heads C_i, r_i' is the cost of protecting cluster heads C_i after T times, p_a^i is the payoff of attacking cluster heads C_i, and p_d^i is the payoff of intrusion detection systems against attacks. It can be seen from the payoff matrix that there is no pure strategy Nash equilibrium in this game, and the intrusion detection system can observe malicious attackers according to its payoff decrease. Besides, the intrusion detection system always tries to determine the cluster head attacked by the attacker and then protect it to maximize its payoff. Therefore, we use the regret minimization algorithm that determines the selection method of that future action according to the degree of regret to determine the players' mixed strategy Nash equilibrium. Thus, the probability of playing strategy d_1 in round T is defined as follows:

$$p(a) = \frac{\text{Regret}_d^T(d_1)}{\sum_{i \in D_d} \text{Regret}_d^T(d_i)}, \tag{5}$$

where D_d is the intrusion detection system's strategy set, $\text{Regret}_d^T(d_1)$ is the regret value of playing strategy d_1, and

$\sum_{i\in D_d}\text{Regret}_d^T(d_i)$ is the cumulative regret value for all strategies.

3.2. Repeated Game. During the process of interaction between the attacker and intrusion detection system, the intrusion detection system can detect attackers' invasion by observing the changes of their payoff. However, the attacker does not have the effect of his current strategy on the future payoff, that is, he only considers the payoff of one interaction; therefore, it is difficult to prevent the attacker in the one-order game. But if the intrusion detection system punishes the attacker, the attacker will have to consider the cost of the penalty brought by the intrusion detection system in the repeated game; and if the punishment cost of attacking exceeds the payoff of attacking, the attacker will be forced to take a nonattack strategy. Thus, the intrusion detection system does not need to implement supervision and then achieve the purpose of maintaining the normal order of the entire network.

In the repeated game, assuming that a_{et} is the strategy adopted by player e in the tth round, the strategy set of player e in the previous Tround is $a_{e1}, a_{e2}, \ldots, a_{eT}$. The total payoff of player e can be expressed as

$$U_e = \sum_{t=1}^{T} \delta^{t-1} u\left(a_{et}, a_{-et}\right), \tag{6}$$

where δ is the discount factor, $\delta \in (0,1)$. The bigger δ is, the more e pays attention to long-term payoff; and the smaller δ is, the more player e pays attention to current payoff. Since the intrusion detection system cannot detect the attacker for the first time, we assume that the detection rate of the intrusion detection system to the attacker is less than 1, $q \in (0,1)$. The probability of an attacker being discovered by an intrusion detection system after k times of attack is $(1-q)^{k-1}q$. The total payoff of the attacker is

$$U_a = \sum_{t=0}^{k} (1-q)^t \delta^t \left(p_a^i - c_i\right). \tag{7}$$

In previous researches on network security protection, once an attacker is captured by the intrusion detection system, the network will delete this node. However, it will affect the whole network and will have no containment effect on the attacker's action. Therefore, this paper designs a severe punishment mechanism to educate captured attackers into regular players. When the attacker is found to be uncooperative at the time slot k, within T penalty cycles, that is, $k+1, k+2, \ldots, k+T$, the attacker's payoff can be defined as

$$U_a^T = \sum_{i=1}^{T}\sum_{t=0}^{k} \frac{1}{k+i} (1-q)^t \delta^t \left(p_a^i - c_i\right). \tag{8}$$

If the node is detected during the second attack, the node will be punished with a period of $2T$, and the total payoff of the attacker in the penalty cycle is

$$U_a^{2T} = \sum_{i=1}^{2T}\sum_{t=0}^{k} \frac{1}{2(k+i)} (1-q)^t \delta^t \left(p_a^i - c_i\right). \tag{9}$$

The loss of attacker in penalty cycle is

$$\Delta U_a^T = \sum_{t=0}^{T} (1-q)^t \delta^t \left(p_a^i - c_i\right) - \sum_{i=1}^{T}\sum_{t=0}^{k} \frac{1}{k+i} (1-q)^t \delta^t \left(p_a^i - c_i\right),$$
$$\Delta U_a^{2T} = \sum_{t=0}^{2T} (1-q)^t \delta^t \left(p_a^i - c_i\right) - \sum_{i=1}^{2T}\sum_{t=0}^{k} \frac{1}{k+i} (1-q)^t \delta^t \left(p_a^i - c_i\right). \tag{10}$$

We regard the loss of the attacker in the penalty cycle as an additional reward to the intrusion detection system. Therefore, the intrusion detection system's payoff can be defined as

$$U_d = \sum_{t=1}^{T} \delta^{t-1} u\left(a_{et}, a_{-et}\right) + \Delta U_a^T, \tag{11}$$

where ΔU_a^T is the loss of attackers in the penalty cycle.

By comparing the attacker's payoffs over the two penalty cycles, it can be seen that the attacker's payoffs decrease with increasing the number of betrayals. Besides, if the number of defections by an attacker exceeds the threshold of the intrusion detection system, the attacker will be eliminated; and the cluster-head node will no longer interact with the attacker.

4. Simulation Experiment

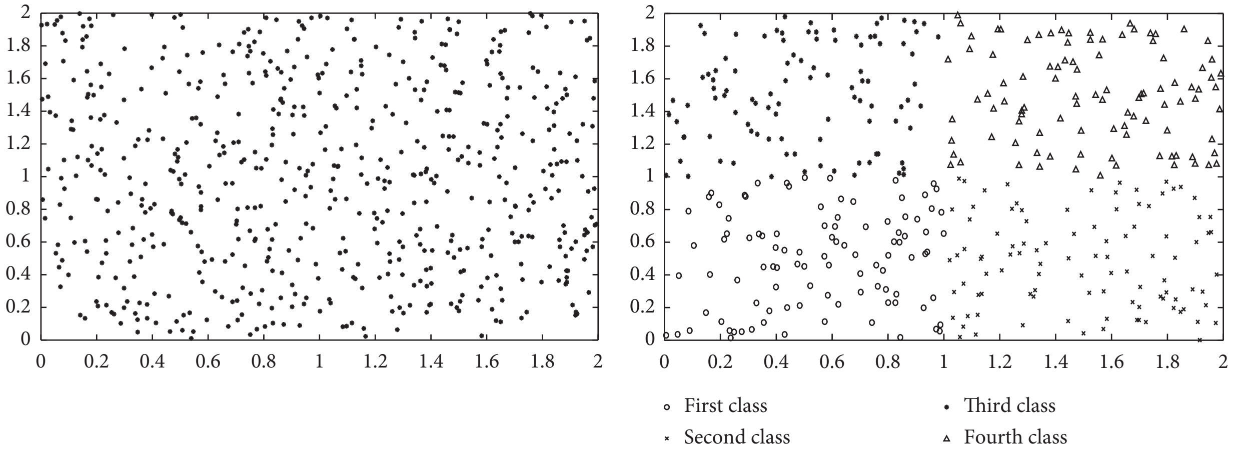
This paper uses Anaconda integrated development tool to verify the intrusion detection scheme based on repeated game. Firstly, we simulate the classification process of KNN algorithm and set four newly added nodes to prove its effectiveness. Secondly, we compare the payoffs of attackers and the intrusion detection systems in penalty cycles and regular interaction cycles to verify the effectiveness of the penalty mechanism. Thirdly, we determine the optimal strategy for each round of interaction between the attacker and intrusion detection system by using the regret minimization algorithm. Finally, we compare the proposed scheme with four interaction strategies, Winner (take the strategy of the winner), ALL-S (remain strategy Scissor), ALL-P (remain strategy Paper), and ALL-R (remain strategy Rock), to prove that the proposed scheme can improve the player's payoff. The experimental parameters are shown in Table 2.

4.1. The Classification Results of KNN. Figure 1 depicts the classification results of the KNN algorithm. Figure 1(a) shows the original distribution of edge device nodes. Figure 1(b) shows the classification results of the KNN algorithm, with each symbol representing a class of edge devices.

Figure 2 analyzes the results of the classification of the newly added nodes, with the newly added nodes marked in blue. For example, in Figure 2(a), the blue node (the newly added node) is classified as a first class.

Table 2: Parameter setting.

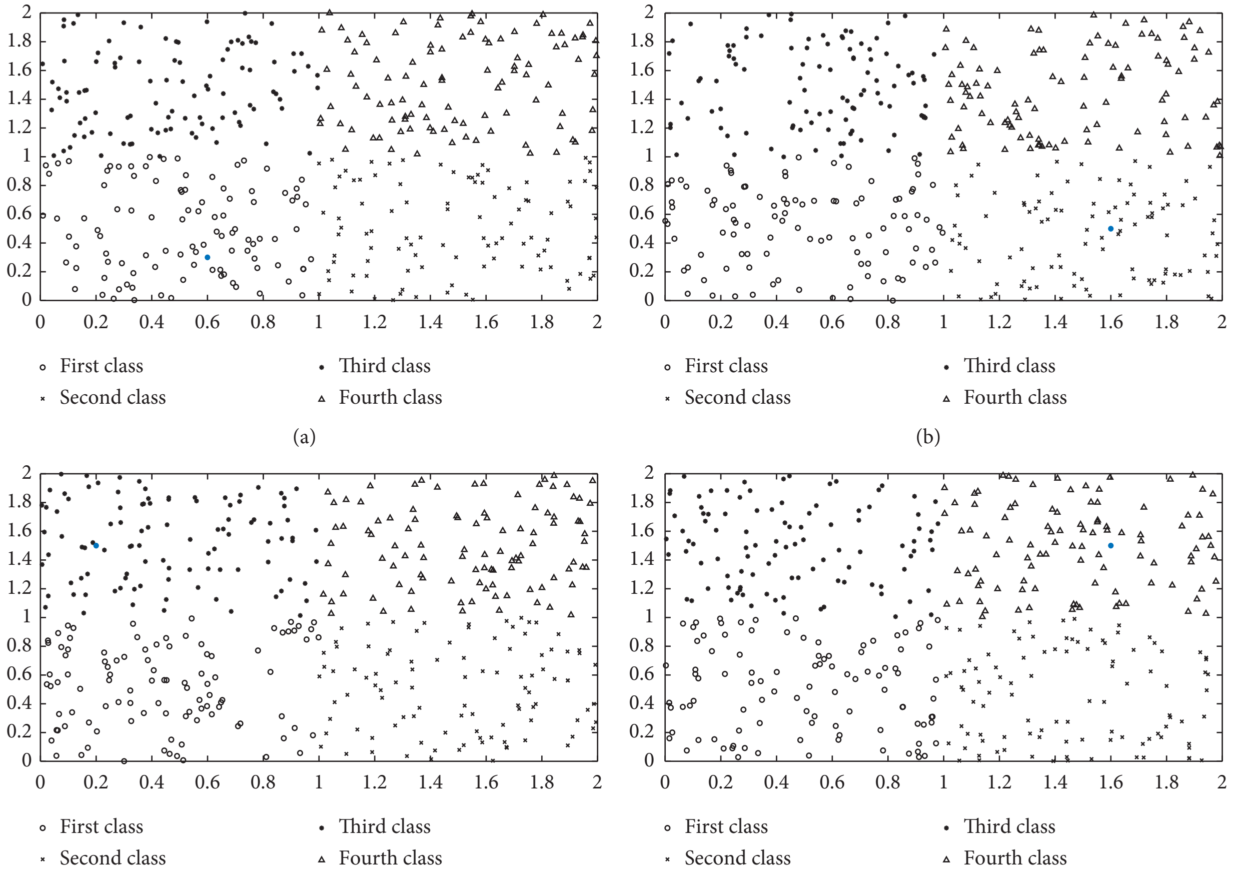
Parameters	p_a^i	p_e	c_i	δ	η	T	q
Value	5	3	3	0.7	1	5	0.6



(a)

(b)

Figure 1: Comparison of classified data. (a) Raw data. (b) Classification results.



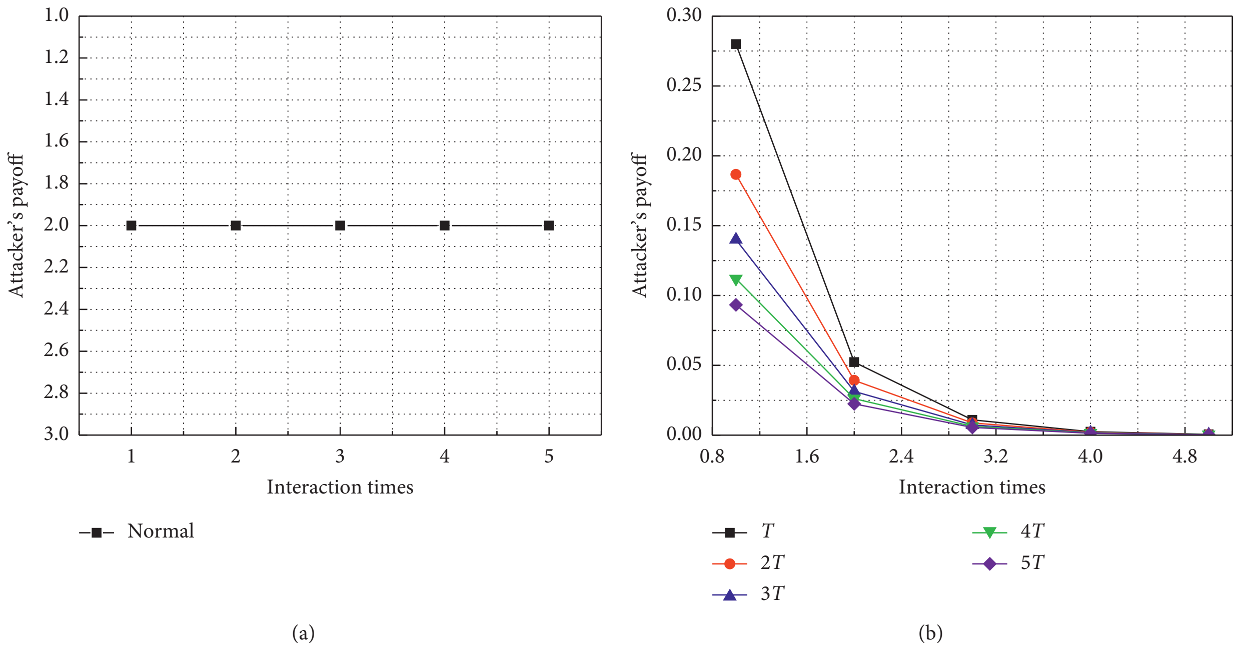
(a)

(b)

(c)

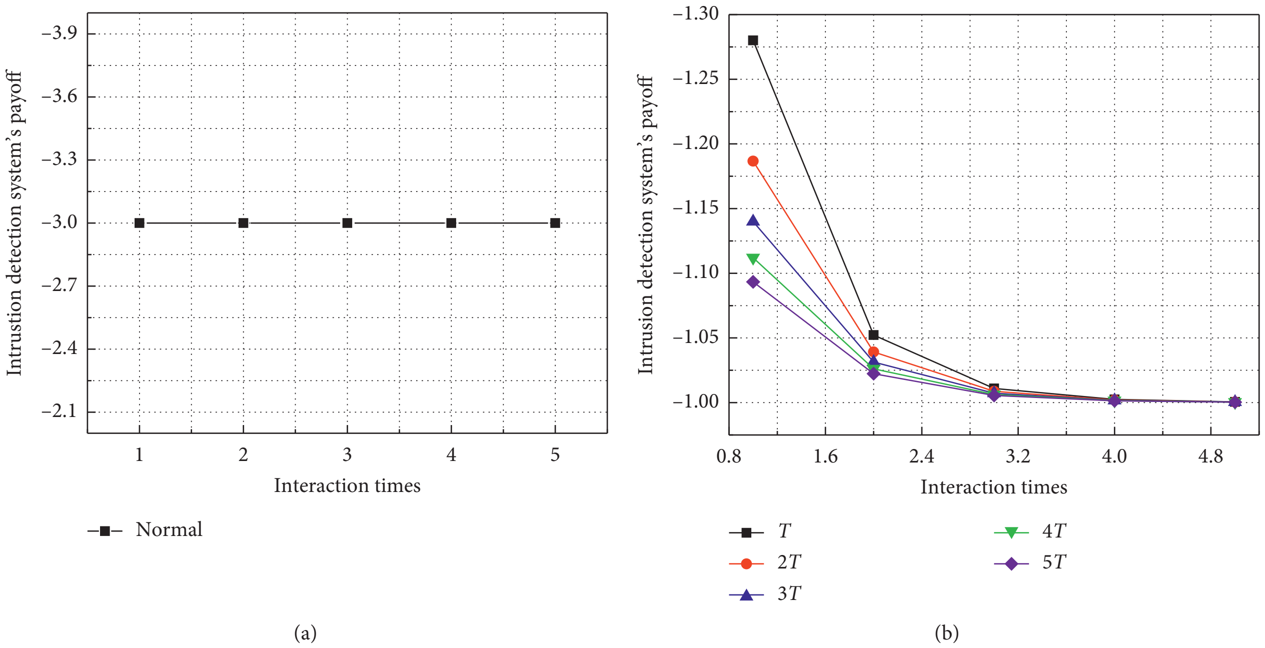
(d)

Figure 2: Classification of newly added data. (a) First class. (b) Second class. (c) Third class. (d) Fourth class.



(a) (b)

Figure 3: The attacker's payoff comparison. (a) The payoffs of regular interactions. (b) Payoff during the penalty period.



(a) (b)

Figure 4: The intrusion detection system's payoff comparison. (a) The payoffs of regular interactions. (b) Payoff during the penalty period.

4.2. The Comparison of the Attacker's Payoff and Intrusion Detection System's Payoff. Figure 3 compares the attackers' payoffs in regular interaction cycles and penalty cycles. As you can see in Figure 3(a), the attacker's payoff does not change during regular interaction cycles, because the intrusion detection system does not play the defensive strategy. Figure 3(b) shows that the attacker's payoff gradually decreased with increasing the number of interactions. In the 4th interaction, the attacker's payoff tends to zero. Besides, the longer the penalty cycle is, the faster the attacker's payoffs will go to zero, and the larger the losses will be. This happened due to the punishment mechanism in this paper. Therefore, for a rational attacker, it must normally interact with the intrusion detection system to maximize its payoff.

Figure 4 compares the intrusion detection system's payoffs in the regular interaction cycle and the penalty cycle. It can be seen from Figure 4(a) that the intrusion detection system's payoff is −3 during the regular interaction cycle.

Table 3: Payoff matrix.

Player A\B	Scissor	Rock	Paper
Scissor	0, 0	−1, 1	1, −1
Rock	1, −1	0, 0	−1, 1
Paper	−1, 1	1, −1	0, 0

Table 4: Regret value of player A.

Iteration number		Player A		Optimal strategy
	Rock	Scissor	Paper	
1	0	2	1	(0, 2/3, 1/3)
2	1	0	2	(1/6, 2/6, 3/6)
3	2	1	0	(1/3, 1/3, 1/3)
4	0	2	1	(3/12, 5/12, 4/12)
5	1	0	2	(4/15, 5/15, 6/15)
6	2	1	0	(1/3, 1/3, 1/3)
7	0	2	1	(6/21, 8/21, 7/21)
8	1	0	2	(7/24, 8/24, 9/24)
9	2	1	0	(1/3, 1/3, 1/3)
10	0	2	1	(9/30, 11/30, 10/30)
Cumulative regret	9	11	10	—

Table 5: Players’ payoff comparison.

Number					Payoff					
	A	B	A	B	A	B	A	B	A	B
1	−1	1	−1	1	−1	1	0	0	1	−1
2	1	−1	−1	1	0	0	0	0	1	−1
3	−1	1	0	0	−1	1	1	−1	−1	1
4	1	−1	0	0	0	0	1	−1	−1	1
5	−1	1	−1	1	−1	1	−1	1	0	0
6	1	−1	−1	1	0	0	−1	1	0	0
7	−1	1	0	0	−1	1	0	0	1	−1
8	1	−1	0	0	0	0	0	0	1	−1
9	−1	1	−1	1	−1	1	1	−1	−1	1
10	1	−1	−1	1	0	0	1	−1	−1	1
Total	0	0	−6	6	−5	5	2	−2	0	0

This is because the attacked intrusion detection system does not play any defective strategy. Figure 4(b) shows that the loss of the intrusion detection system decreases with increasing the number of penalty cycles; and the payoff of the intrusion detection system is the lowest when the penalty period is 5. To sum up, the proposed scheme can reduce the loss of intrusion detection systems when attackers launch attacks.

4.3. Application of Regret Minimization Algorithm in Rock-Paper-Scissors Game. Table 3 defines the payoff matrix of two players in the rock-paper-scissors game. In this table, the rows represent the strategy of player A, the columns represent the strategy of player B, the first element in the tuple (0, 0) represents the payoff of player A, and the second element represents the payoff of player B.

Table 4 analyzes how player *A* determines its optimal strategy based on the regret minimization algorithm. For example, in the first round, player A and player B choose Rock and Paper, respectively, and then player A’s regret values when playing Scissor, Rock, and Paper are 0, 2, and 1, respectively; thus the probabilities of player playing Rock, Scissor, and Paper are 0, 2/3, and 1/3, respectively. Similarly, we can obtain the optimal strategy of player A in each round.

4.4. The Payoff Comparison between Player A and Player B. Table 5 compares the payoffs of player A and player B when player A adopts five strategies: regret minimization strategy (Regret), ALL-R, ALL-P, ALL-S, and Winner, while player *B* adopts a regret minimization strategy. As can be seen from Table 5, when and only if player A adopts ALL-P, player B adopts Regret to obtain a lower payoff than player A, but the difference in payoff between player A and player B is small. However, under several other strategies, player B obtains the highest payoff by taking Regret. This is because player B maximizes the probability of the strategy with the maximum regret value. The payoff change curves of players A and B are shown in Figure 5. In this figure, the sharp increase and

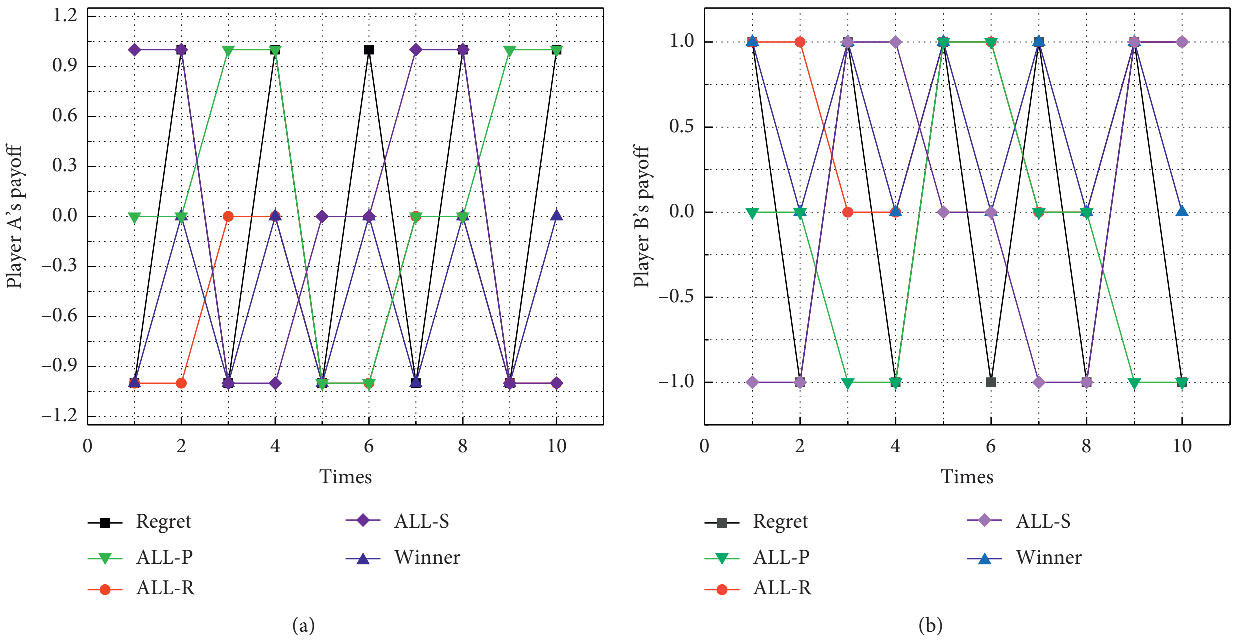


Figure 5: Players' payoff comparison. (a) The payoff of player A. (b) The payoff of player B.

decrease in the payoffs of player A and player B are due to the adjustment of both players' strategies.

5. Conclusion

Designing an efficient and safe protection scheme is the key to promoting the application of the system. This paper proposes a security protection scheme based on repeated game. In this scheme, the intrusion detection system detects the malicious attackers by observing its payoff change and punishes the attackers who adopt malicious strategy severely to educate the attackers to take good action. The experimental results show that the proposed scheme can effectively defend against the attackers.

In future research studies, we will continue to explore new methods to determine the player's optimal strategy in the finite model.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (NSFC) under Grant no. 61872205, the Shandong Provincial Natural Science Foundation under Grant no. ZR2019MF018, and the Source Innovation Program of Qingdao under Grant no. 18-2-2-56-jch.

References

[1] N. Chen, T. Qiu, X. Zhou, K. Li, and M. Atiquzzaman, "An intelligent robust networking mechanism for the internet of things," *IEEE Communications Magazine*, vol. 57, no. 11, pp. 91–95, 2019.

[2] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart internet of things systems: a consideration from a privacy perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.

[3] H. Xia, L. Li, X. Cheng, C. Liu, and T. Qiu, "A dynamic virus propagation model based on social attributes in city IoTs," *IEEE Internet of Things Journal*, 2020.

[4] M. A. Hatef, V. Shaker, M. Reza Jabbarpour, J. Jung, and H. Zarrabi, "HIDCC: a hybrid intrusion detection approach in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 3, p. e4171, 2018.

[5] S. Venkatraman and B. Surendiran, "Adaptive hybrid intrusion detection system for crowd sourced multimedia internet of things systems," *Multimedia Tools and Applications*, vol. 79, no. 5-6, pp. 3993–4010, 2020.

[6] K. K. R. Amrita, "A hybrid intrusion detection system: integrating hybrid feature selection approach with heterogeneous ensemble of intelligent classifiers," *International Journal of Network Security*, vol. 20, no. 1, pp. 41–55, 2018.

[7] T. Qiu, J. Liu, W. Si, and D. O. Wu, "Robustness optimization scheme with multi-population Co-evolution for scale-free wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1028–1042, 2019.

[8] H. Xia, L. Li, X. Cheng, X. Cheng, and T. Qiu, "Modeling and analysis botnet propagation in social internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 8, 2020.

[9] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[10] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: a survey," 2019, https://arxiv.org/abs/1901.03407.

[11] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the 31st*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6479–6488, Salt Lake City, UT, USA, June 2018.

[12] D. Kwon, H. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, vol. 22, pp. 1–13, 2017.

[13] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrão, and M. L. Proença, "Network anomaly detection system using genetic algorithm and fuzzy logic," *Expert Systems with Applications*, vol. 92, pp. 390–402, 2018.

[14] X. Kong, X. Song, F. Xia, H. Guo, J. Wang, and A. Tolba, "LoTAD: long-term traffic anomaly detection based on crowdsourced bus trajectory data," *World Wide Web*, vol. 21, no. 3, pp. 825–847, 2018.

[15] H. Xia, S.-S. Zhang, Y. Li, Z.-K. Pan, X. Peng, and X.-Z. Cheng, "An attack-resistant trust inference model for securing routing in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 7108–7120, 2019.

[16] X. Zheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Follow but no track: privacy preserved profile publishing in cyber-physical social systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1868–1878, 2017.

[17] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang, "Specguard: spectrum misuse detection in dynamic spectrum access systems," *IEEE Transactions on Mobile Computing*, vol. 17, no. 12, pp. 2925–2938, 2018.

[18] H. A. Seven, H. A. Nguyen, S. Nadi, T. N. Nguyen, and M. Mezini, "Investigating next steps in static API-misuse detection," in *Proceedings of the 16th International Conference on Mining Software Repositories*, pp. 265–275, Montreal, Canada, May 2019.

[19] S. Amann, H. A. Nguyen, S. Nadi, T. N. Nguyen, and M. Mezini, "A systematic evaluation of static API-misuse detectors," *IEEE Transactions on Software Engineering*, vol. 45, no. 12, pp. 1170–1188, 2018.

[20] T. Qiu, B. Li, X. Zhou, H. Song, I. Lee, and J. Lloret, "A novel shortcut addition algorithm with particle swarm for multisink internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3566–3577, 2020.

[21] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Computer Networks*, vol. 136, pp. 37–50, 2018.

[22] H. Li, K. Ota, and M. Dong, "Deep reinforcement scheduling for mobile crowdsensing in fog computing," *ACM Transactions on Internet Technology*, vol. 19, no. 2, pp. 1–18, 2019.

[23] H. Zhang, J. Yu, C. Tian et al., "Efficient and secure outsourcing scheme for RSA decryption in internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6868–6881, 2020.

[24] H. Zhang, J. Yu, C. Tian, G. Xu, P. Gao, and J. Lin, "Practical and secure outsourcing algorithms for solving quadratic congruences in internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2968–2981, 2020.

[25] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, and W. Lv, "Edge computing security: state of the art and challenges," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1608–1631, 2019.

[26] D. Yu, Y. Zou, J. Yu et al., "Implementing abstract MAC layer in dynamic networks," *IEEE Transactions on Mobile Computing*, 2020.

[27] D. Yu, Y. Zou, J. Yu et al., "Stable local broadcast in multihop wireless networks under SINR," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1278–1291, 2018.

[28] F. Li, D. Yu, H. Yang, J. Yu, H. Karl, and X. Cheng, "Multi-armed-bandit-based spectrum scheduling algorithms in wireless networks: a survey," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 24–30, 2020.