Research Article

Network Security Defense Decision-Making Method Based on Stochastic Game and Deep Reinforcement Learning

Zenan Wu ¹, Liqin Tian ^{1,2}, Yan Wang ³, Jianfei Xie ², Yuquan Du ²,
and Yi Zhang ²

¹School of Computer, Qinghai Normal University, Xining 810000, China

²School of Computer, North China Institute of Science and Technology, Beijing 101601, China

³Department of Robot, Ningbo University of Technology, Ningbo 315211, China

Correspondence should be addressed to Liqin Tian; 18511465255@163.com

Received 2 August 2021; Revised 2 November 2021; Accepted 20 November 2021; Published 16 December 2021

Academic Editor: Yanhui Guo

Copyright © 2021 Zenan Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the existing network attack and defense stochastic game models, most of them are based on the assumption of complete information, which causes the problem of poor applicability of the model. Based on the actual modeling requirements of the network attack and defense process, a network defense decision-making model combining incomplete information stochastic game and deep reinforcement learning is proposed. This model regards the incomplete information of the attacker and the defender as the defender's uncertainty about the attacker's type and uses the Double Deep Q-Network algorithm to solve the problem of the difficulty of determining the network state transition probability, so that the network system can dynamically adjust the defense strategy. Finally, a simulation experiment was performed on the proposed model. The results show that, under the same experimental conditions, the proposed method in this paper has a better convergence speed than other methods in solving the defense equilibrium strategy. This model is a fusion of traditional methods and artificial intelligence technology and provides new research ideas for the application of artificial intelligence in the field of cyberspace security.

1. Introduction

In recent years, with the rapid development of information technology, network attacks have increased. Many new attack methods have been presented. Information network security has always been a hot issue [1], especially in some traditional networks. In order to reduce the complexity of the network and make the network equipment show strong homogeneity, network equipment has to be more vulnerable to network attacks. Once one network node is destroyed, the entire network system will be paralyzed. When a malicious attacker launches an attack by exploiting the loopholes behind the network equipment, the normal operation of the network will be disturbed. Besides, the leakage of major information will be caused. If the situation gets even worse, the security of the entire network system will be endangered [2]. Due to the complexity of the network system and the concealment and stochasticity of the attack means, it is hard for existing network

defense technology to meet the security requirements of the network system, which makes it harder for the defender of the network system to guarantee the absolute security of the system. Therefore, there is a need for a new technology that can analyze network attack and defense events, so that network system defenders can implement dynamic and adaptive adjustment of defense strategies [3].

Primarily, due to the huge amount of similar characteristics of game theory and network offensive and defensive events, such as the antagonism of participants' goals, non-cooperation of strategies, and behavioral dependence, relevant research on game theory in network information security is increasingly emerging [4, 5]. Furthermore, the stochastic game is a dynamic game with state transition and is composed of a series of stages. It performs well because of describing multiple states. Therefore, stochastic game has quickly become a hot spot in the current research on network offense and defense [6]. Wang et al. [7] proposed a method for

quantitative analysis of network security based on stochastic game model, which analyzes and evaluates the security of the target network based on the model generated by the simulation. Fu et al. [8] broke through the difficulties in the process of network offense and defense from the perspective of stochastic games. They used the quantification of the benefits of both offense and defense to propose a new selection algorithm to cope with the difficulty of responding to changes in attack intent and strategy in the process of network offense and defense. Huang and Zhang [9] aimed at the defect that the traditional deterministic game model could not accurately describe the offensive and defensive process in the real network environment and proposed a security defense strategy selection model based on the stochastic offensive and defensive evolutionary game model. The security defense strategy uses Gaussian white noise and stability judgment theorem of stochastic differential equations has made a breakthrough in the direction of the offensive and defensive strategies. Hu et al. [10] embedded noncooperative signal game theory in network attack and defense and simulated the dynamic network attack and defense confrontation process with the aid of the dynamic attenuation effect of network deception signals. New research ideas were proposed in the direction of network security active defense. Wei et al. [11] applied game theory to the maintenance of power grid security, developed a new model framework through stochastic game theory on the interaction between offense and defense, and introduced new algorithms to enhance the power grid's ability to defend against attacks. Lei et al. [12] proposed a new mobile target defense strategy generation method based on incomplete information Markov game theory in view of the fact that the classic game theory and the complete information hypothesis cannot describe the mobile target defense confrontation problem well. In addition, with the development of emerging technologies such as artificial intelligence and machine learning in recent years, more and more intelligent algorithms have been applied to the field of network security [13, 14]. In order to enable the network to provide people with efficient and fast services, Hua et al. [15] integrated artificial intelligence technology into the field of network security and proposed a system detection algorithm based on the concept of artificial intelligence. This work has played a very supportive role for artificial intelligence technology to perform security inspections on network systems. In order to analyze the influence of bounded rationality on the stochastic game of network offense and defense, Zhang and Liu [16] aimed at the problem of state explosion when the number of network nodes increases; an offensive graph and a defensive graph have been designed to compress the state space and extract the network state and defense strategy. On this basis, the introduction of intelligent learning algorithms and the design of defense decision-making algorithms with online learning capabilities would help to select the optimal defense strategy with the greatest benefit from the set of candidate strategies. However, this method is similar to the Q-learning algorithm and is prone to "overestimation."

Although the above studies provide solutions for the analysis of network attack and defense events, there are still some shortcomings: (1) Most studies are based on the

assumption of complete information. However, in a real network attack and defense event, due to the concealment of the attacker, the defender cannot fully grasp the relevant information of the attacker. (2) The income functions of the above literatures are all based on the known transfer model. But, in many cases, the defender cannot grasp the probability of system state transition. Therefore, the above two points make the applicability of the model proposed in the abovementioned literature poor.

In response to the above problems and to improve the applicability of the stochastic game model in the analysis of network offensive and defensive events, this paper proposes a defense strategy selection model based on incomplete information stochastic game. In the meantime, we draw on the idea of reinforcement learning using the Double Deep Q-Network algorithm to conduct game analysis on the stochastic game model. Thus, the defender's income can be dynamically updated, and the defense strategy can be adaptively adjusted. It is not necessary to set the system state transition probability in advance to obtain the Nash equilibrium of the two sides of the game. Lastly, the validity of the proposed model is verified through experiments.

The main contributions of this article are as follows:

- (1) Improve the existing network offensive and defensive stochastic game model and regard the incomplete information between people in the game as the defender's uncertainty about the attacker's type, so that the model can meet the real network offensive and defensive scenarios.
- (2) The deep reinforcement learning algorithm, Double DQN, is introduced into the model, so that the defender's income can be learned and updated online, and the accuracy of the model's solution to the defense strategy is improved.

This paper is organized as follows: Section 1 introduces the background and related work. Stochastic game model with incomplete information is presented in Section 2. Deep reinforcement learning method is discussed in Section 3 and experimental analysis in Section 4. In the end, comparative analysis of related work is discussed in Section 5 and the conclusion is given in Section 6.

2. Stochastic Game Model with Incomplete Information

Due to factors such as the network environment and network entities, network attack and defense is a complicated and stochastic process. The attacker and the defender have a relationship of target opposition and behavior dependence. Therefore, the network attack and defense process can be described as a stochastic game process between the attacker and the defender. In addition, for their own interests, both the offensive and defensive parties will always hide their own information from each other. Therefore, the network offensive and defensive process in the actual environment is a stochastic game process with incomplete information [17].

2.1. Discrete Processing of Network Attack and Defense Process. In order to facilitate the modeling and analysis of the network attack and defense process, we firstly discretize the network attack and defense process [18]. The whole process is regarded as a series of time slices $\{T_1, T_2, T_3, \dots, T_n\}$; each time slice contains only one network state, and each time slice is an offensive and defensive game process. The process can be described in Figure 1.

With the occurrence of cyberattacks, the network system transfers from one state to another under the interaction of the entities, as shown in Figure 2. The state transition of the network system is stochastic. In addition to being affected by the actions of malicious users, it is also affected by some other complex factors in the network. Our research goal is to find the defense strategy that the defender can obtain higher returns in the network attack and defense stochastic game.

2.2. Network Attack and Defense Stochastic Game Model. Due to the stochasticity and dynamics of the network state transition, the state transition of the network system is constantly changing in the long term. In the meanwhile, the next transition state is only related to the current state. In conclusion, the transition of the network state can be regarded as a Markov decision process: $P(s_{t+1}|S_0:t, a_0:t, d_0:t) = P(s_{t+1}|S_t, a_t, d_t)$; in the short term, the state transition of the network system is a fixed value [19].

Definition 1. The incomplete information-attack and defense stochastic game model is a 9-tuple: $II - \text{ADSGM} = (N, S, A, D, \vartheta, P_A, \pi, R, U)$, where

- (1) $N = \{\text{Attacker, Defender}\}$ is the collection of the attacker and the defender, that is, the player in the game
- (2) $S = \{s_1, s_2, \dots, s_n\}$ is the state set of the network system, that is, the set of stochastic game states
- (3) $A = \{A_1, A_2, \dots, A_n\}$ is the behavior set of the attacker, where $A_i = (a_1, a_2, \dots, a_j)$ is the behavior strategy set of the attacker in the system state s_i

- (4) $D = \{D_1, D_2, \dots, D_n\}$ is the behavior set of the defender, where $D_i = (d_1, d_2, \dots, d_j)$ is the behavior strategy set of the defender in system state s_i
- (5) $\vartheta = \{\theta_1, \theta_2, \dots, \theta_m\}$ is the set of attacker types
- (6) $P_A = \{p_A(s_i, \theta_1), p_A(s_i, \theta_2), \dots, p_A(s_i, \theta_m)\}$ represents the set of probabilistic judgments of the defender on the type of attacker when the system is in s_i
- (7) $\pi = (\pi_a, \pi_d)$ represents a set of offensive and defensive strategies, where $\pi_a(s_i, \theta_j) = (\sigma_a(s_i, a_1, \theta_j), \dots, \sigma_a(s_i, a_m, \theta_j))$ represents the strategy of the θ_j -shaped attacker in the system state, and $\sigma_a(s_i, a_i, \theta_j)$ is the probability of the attacker's choice of behavior a_i ; similarly, $\pi_d(s_i)$ represents the defense strategy of the defender when the system is in s_i , and $\sigma_d(s_i, d_j)$ is the probability of choosing behavior d_j for the defender
- (8) $R_{a,d}(s_i, a, d, \theta_j)$ means that, in the system state s_i and the attacker type θ_j , the offensive and defensive parties take an immediate return of action (a, d)
- (9) $U = (Q, V)$ is the set of revenue functions for both offensive and defensive parties, where $Q = (Q_a, Q_d)$ represents the state-behavior revenue function set of both offensive and defensive parties, and $Q_d = (s_i, a, d, \theta_j)$ represents the revenue function of the defender after the offensive and defensive parties take actions when the system state is s_i and the attacker type is θ_j ; $V = (V_a, V_d)$ represents the state-strategy value revenue function set of the offensive and defensive parties; $V_d(s_i, \pi_a(s_i, \theta_j), \pi_d(s_i, \theta_j))$ represents the revenue function of the defender after the offensive and defensive parties adopt the strategy $(\pi_a(s_i, \theta_j), \pi_d(s_i))$ when the attacker type is θ_j in the system state s_i

According to the analysis of the network attack and defense process and the definition of the above model, the defender's profit function is expected to accumulate. Therefore, the defender's state-strategy value profit function V_d is expressed as

$$V_d(s_i, \pi_a(s_i, \theta_j), \pi_d(s_i, \theta_j)) = \sum_{a_i \in A_i} \sigma_a(s_i, a_i, \theta_j) \sum_{d_i \in D_i} \sigma_d(s_i, d_i) Q_d(s_i, a, d, \theta_j). \quad (1)$$

3. Deep Reinforcement Learning and Bayesian Equilibrium Solution

3.1. Bayesian Nash Equilibrium in the Process of Network Attack and Defense. Since network offense and defense can be regarded as a stochastic game process with incomplete information, the equilibrium solution is Bayesian Nash equilibrium. That is, both offense and defense can no longer

unilaterally change their strategies to improve their own profits.

Definition 2. Bayesian Nash equilibrium of network offense and defense. There are (1) all attack strategies $\pi_a(s_i, \theta_j)$ of the attacker and (2) all defense strategies $\pi_d(s_i)$ of the defender, satisfying

FIGURE 1: Time slice of network attack and defense game. In each time slice, both the offensive and defensive parties check the current state of the system and select strategies based on the state.

FIGURE 2: Network state transition. The state of the system changes under the action of offensive behavior and defensive behavior.

$$\begin{aligned}
 & V_a(s_i, \pi_a^*(s_i, \theta_j), \pi_d^*(s_i), \theta_j) \geq V_a(s_i, \pi_a(s_i, \theta_j), \pi_d^*(s_i), \theta_j), \\
 & \sum_{j=1}^m p_A(s_i, \theta_j) V_d(s_i, \pi_a^*(s_i, \theta_j), \pi_d^*(s_i), \theta_j) \geq \sum_{j=1}^m p_A(s_i, \theta_j) V_d(s_i, \pi_a(s_i, \theta_j), \pi_d(s_i), \theta_j).
 \end{aligned} \tag{2}$$

Then strategy $(\pi_a^*(s_i, \theta_j), \pi_d^*(s_i))$ is a Bayesian Nash equilibrium in the state of the system, and the equilibrium return at this time is denoted as Q^* .

The equilibrium solution of II-ADSGM mentioned in this paper is the set of Bayesian Nash equilibrium solutions for each state of the system. This problem can be regarded as a secondary planning problem:

$$f: (Q_a(s_i, \theta_1), \dots, Q_a(s_i, \theta_n), Q_d(s_i, \theta_1), \dots, Q_d(s_i, \theta_n), P_A) \longrightarrow (\pi_a^*(s_i, \theta_j), \pi_d^*(s_i)). \tag{3}$$

In actual network offensive and defensive incidents, the decisions of both sides are often continuous. Therefore, the decision will affect not only current but also future benefits

of both parties. As in the above definition, the benefits obtained by the offensive and defensive parties in the game process should include current benefits and future benefits,

and the benefits change dynamically with the strategy. Therefore, in the network attack and defense stochastic game

type, the revenue function Q of the offense and defense is defined as

$$Q(s, a, d, \theta_j) = R_{a,d}(s, a, d, \theta_j) + \gamma \sum_{s' \in S} (p(s'|s, a, d) V(s', \pi_a(s', \theta_j), \pi_d(s'), \theta_j)). \quad (4)$$

$R_{a,d}(s, a, d, \theta_j)$ represents the current reward, $\gamma \sum_{s' \in S} p(s'|s, a, d) V(s', \pi_a(s', \theta_j), \pi_d(s'), \theta_j)$ represents the future reward, and γ ($0 < \gamma < 1$) is the discount factor. Obviously, when γ is grater, the revenue function is more affected by the future return, and when γ is less, the revenue function is more affected by the current return; $p(s'|s, a, d)$ represents the probability of the system transitioning from state to state under the influence of action (a, d) . Because in the actual network attack and defense process the system state transition probability is dynamically changing, it is difficult to determine the value of parameter $p(s'|s, a, d)$, which is the follow-up solving the Nash equilibrium of the system has brought great inconvenience. In most of the existing studies, the value of parameter $p(s'|s, a, d)$ is set in advance to facilitate subsequent calculations, which is obviously not in line with the actual situation.

In our research, we aim to solve the Bayesian Nash equilibrium of the system when the unknown parameter $p(s'|s, a, d)$ changes dynamically. In constructed offensive and defensive stochastic game model, the equilibrium income Q_* needs to meet the conditions that can be updated online with the offensive and defensive process. Therefore, in terms of network system security requirements, defenders are required to adapt to their defense strategies.

3.2. Q-Learning Algorithm. Q-learning is the basic algorithm in the reinforcement learning [20, 21], where the revenue function $Q(S_t, A_t)$ represents the revenue expectation that the agent can obtain by taking behavior A in state S at time t . Q is related to returns and the network environment, and the specific formula is as follows:

$$Q(S_t, A_t) = (1 - \alpha)Q(S_t, A_t) + \alpha \left(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) \right). \quad (5)$$

α is the learning rate, which could dynamically adjust the income. After a dynamic adjustment process, the solution of income Q^* would not depend on the system state transition probability. It would also make up for the deficiencies of the existing model.

The offensive and defensive confrontation between network entities is a complicated process. In view of defenders' strategy selection problem, many existing studies have simplified the offensive and defensive process as necessary [22–24]. Considering that, the behaviors of various entities affect each other, and the transition of the system state caused by the interactive behaviors would provide a reference for the entities to select behaviors again. Therefore, the behavior learning mechanism of entities in a network attack and defense event is shown in Figure 3.

3.3. Solving State Transition Probability Parameters Based on Double Deep Q-Network Algorithm. Although the Q-learning algorithm is widely used in network attack and defense event analysis [25], it also has some shortcomings: because the Q-learning algorithm uses a Q-table to store the Q-value of each state-behavior, when the state and behavior spaces are discrete and the dimensionality is not high, the algorithm is effective. If the state and behavior space is continuous and the dimensionality is higher, the resulting behavior space and state space would be too large. Under these circumstances, it is difficult to solve the value of all state actions. Thus Q-learning cannot maintain such a large Q-table in memory.

In response to the above problems, some researchers have proposed using a model to represent the relationship between the state and the action to the value function. Deep Q-Network (DQN) is an algorithm formed by deep learning and reinforcement learning [26]. Compared with the Q-learning algorithm, it uses a neural network to approximate the behavior value function, transforms the Q-table update into a function fitting problem, and then fits a function to replace the Q-value generated in the Q-table, as shown in formula (6). In this way, the similar state can get the similar output behavior. In addition, the DQN algorithm also introduces a target value Q-Network that is independent and slower than the current value of Q-Network, as well as a playback memory unit. The structure of the DQN algorithm is shown in Figure 4. Therefore, the DQN algorithm has a better effect on the extraction of complex features than the Q-learning algorithm

$$Q(S_t, A_t) \approx f(S_t, A, \omega) \approx Q(S_t, A_t) + \alpha \left(R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \omega^-) - Q(S_t, A_t; \omega) \right). \quad (6)$$

Although the DQN algorithm is more suitable for the analysis of network attack and defense events than the Q-learning algorithm, it also has shortcomings: the DQN

algorithm cannot overcome the inherent shortcomings of Q-learning itself, overestimation; that is, the estimated value function is larger than the true value function, and its root is

FIGURE 3: The learning mechanism of Q-learning algorithm in network attack and defense events. The attacker/defender must consider not only the network environment but also the behavior of the other party when learning or making decisions.

FIGURE 4: Deep Q-Network algorithm flowchart in the network attack and defense environment.

mainly in the maximization operation in Q-learning. From formula (6), it can be concluded that the target in the action selection is $R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \omega^-) - Q(S_t, A_t; \omega)$, where the max operation makes the estimated value function larger than the true value of the value function.

However, considering the real network offensive and defensive events, the strategy of offensive and defensive parties does not always choose the action that maximizes the

Q-value in a given state. In general, real strategy is a stochastic strategy; thus, the direct selection of the Q-value with the largest action for the target value will often cause the target value to be greater than the true value.

In order to solve this shortcoming, Hasselt proposed the Double DQN method [27], the definition of which is to use different value functions for the selection of actions and the evaluation of actions. The calculation formula is

$$Q(S_t, A_t) \approx f(S_t, A_t, \omega, \omega^-) \approx Q(S_t, A_t) + \alpha \left(R_{t+1} + \gamma Q(S_{t+1}, \arg \max_a Q(S_{t+1}, a; \omega^-) - Q(S_t, A_t; \omega) \right). \quad (7)$$

In addition, because the revenue $Q(S_t, A_t)$ of the traditional DQN algorithm is only related to the environment and behavior and only one type of participants is involved in the algorithm, in a cyberattack event, there are two types of participants: attacker and defender. Therefore, the revenue

function in the Double DQN algorithm needs to be expanded from one type of participants to two types of participants, and formula (7) needs to be improved. If we take the benefit of the defender as an example, the improved benefit would be

$$Q_d(S_t, a, d) \approx f_d(S_t, a, d, \omega, \omega^-) \approx Q_d(S_t, a, d) + \alpha \left(R_{t+1} + \gamma Q(S_{t+1}, \arg \max_d Q(S_{t+1}, d; \omega^-) - Q_d(S_t, a, d; \omega) \right). \quad (8)$$

In this way, using formula (8), the value of the model's state-action gain Q no longer depends on parameter

$p(s'|s, a, d)$. Then, according to the learning rate α , Q achieves the equilibrium gain Q^* through the learning mechanism.

FIGURE 5: Network system topology.

Input: II – ADSGM; learning rate α ; reward discount factor γ ; exploration probability ε ; convergence accuracy δ ; stable duration κ ;
Output: optimal defense strategy π_d^* .

Begin

- (1) Initialization: $S = \{s_1, s_2, \dots, s_n\}; \mathcal{V} = \{\theta_1, \theta_2, \dots, \theta_n\}; P_A = \{p_A(s_i, \theta_1), p_A(s_i, \theta_2), \dots, p_A(s_i, \theta_m)\},$
 $A_i = (a_1, a_2, \dots, a_j); D = \{D_1, D_2, \dots, D_n\}; Q = (Q_a, Q_d)$
- (2) Solve Bayesian Nash equilibrium: $(\pi_a^*(s, \theta_1), \dots, \pi_a^*(s, \theta_n)) = \text{lebg-plex}(Q(s), P_A), s \in S$
- (3) Network defense strategy revenue function: $V_d(s_i, \pi_a(s_i, \theta_j), \pi_d(s_i, \theta_j)) = \sum_{a_i \in A_i} \sigma_a(s_i, a_i, \theta_j) \sum_{d_i \in D_i} \sigma_d(s_i, d_i) Q_d(s_i, a, d, \theta_j)$
- (4) Get current network status: $s = \text{get}(E)$
- (5) repeat:
- (6) Select defensive actions through ε -greedy algorithm: $d = \pi^\varepsilon(s)$
- (7) Output d
- (8) Get a new network status: $s' = \text{get}(E)$
- (9) Update and learn Q according to the phased results: $Q_d(S_t, a, d) \approx f_d(S_t, a, d, \omega, \omega^-) \approx Q_d(S_t, a, d) + \alpha(R_{t+1} + \gamma Q(S_{t+1}, \arg \max Q(S_{t+1}, d; \omega^-) - Q_d(S_t, a, d; \omega))$
- (10) Update Bayesian Nash Equilibrium: $(\pi_a^*(s, \theta_1), \dots, \pi_a^*(s, \theta_n)) = \text{lebg-plex}(Q(s), P_A), s \in S$
- (11) $V_d(s_i, \pi_a(s_i, \theta_j), \pi_d(s_i, \theta_j)) = \sum_{a_i \in A_i} \sigma_a(s_i, a_i, \theta_j) \sum_{d_i \in D_i} \sigma_d(s_i, d_i) Q_d(s_i, a, d, \theta_j)$
- (12) $s = s'$
- (13) Until $|\pi_d^{t-i^*} - \pi_d^{t-i-1^*}| \leq \delta, \forall i \in [0, \kappa]$
- (14) Output π_d^*
- (15) End

ALGORITHM 1: Cyber adaptive defense countermeasures algorithm.

In addition, this paper uses the ε -greedy algorithm to solve the exploration and utilization problem in the Double DQN algorithm; that is, the algorithm stochastically selects the behavior of the next time slice with the probability of ε and uses the probability of $1 - \varepsilon$ to obtain the Nash equilibrium strategy.

3.4. Cyber Adaptive Defense Countermeasure Algorithm. For each time slice in a cyberattack event, the algorithm uses II – ADSGM to model and analyze the offensive and

defensive process, solves the Bayesian Nash equilibrium according to the participants' Q , and makes defense decisions and then uses the improved Double DQN algorithm to conduct the offensive and defensive confrontation process. Online learning is performed and Q is updated. The specific method is as follows.

Let $|S|$ be the number of states of the network system, let $|A|$ be the number of measures that the attacker can implement in each state, and let $|D|$ be the number of measures that the defender can implement in each state. The space

FIGURE 6: Defender's strategy selection under different parameter settings. (a) Experimental parameter 1. (b) Experimental parameter 2. (c) Experimental parameter 3.

complexity of Algorithm 1 is mainly concentrated on the storage of $R_d(s, d)$, $\pi_a^*(s, \theta_i)$, $\pi_d^*(s, \theta_i)$, and $Q_d(S_t, a, d)$; thus, the space complexity is $O(|S|^2|D| + |S|(|A| + |D| + |A||D|))$. The time complexity of Algorithm 1 is mainly focused on the update of $\pi^*(s)$ after the strategy is selected. We use the Lebg-plex algorithm to calculate it. The average time complexity of the Lebg-plex algorithm is $O((\max(|A|, |D|))^3)$.

4. Simulation Experiment and Analysis

In order to verify the correctness and rationality of the model proposed in this paper and the method of solving the equilibrium, the network environment used in the simulation experiment draws on the typical experimental network constructed by [25], and the topology is shown in Figure 5.

The experimental data used in this paper comes from the offensive and defensive behavior database of MIT Lincoln Laboratory. In the case of known system state transition probability, the detailed process of solving the Nash equilibrium strategy is shown in Appendix B. We use Python 2.7 to implement the algorithm for selecting defense strategies under the condition that the system state transition probability is unknown. The performance of the algorithm is shown in Figure 6.

From the Figure 6, we find that, under the parameter settings shown in Table 1, the probability values of the two different strategies selected by the defender can converge to the Bayesian Nash equilibrium, which is consistent with the calculation results in Table 2, indicating that this article improves the accuracy of the model. At the same time, we found that the convergence

TABLE 1: Experimental parameters.

No.	α	γ	ε
1	0.2	0.9	0.1
2	0.4	0.6	0.2
3	0.6	0.3	0.3

FIGURE 7: Changes in defense strategies of two different algorithms.

speeds of the defense decision values corresponding to the three groups of different parameters are different. It can be clearly seen from the figure that the third group reaches equilibrium when the number of defenses is about 140 times, which is significantly faster than the convergence speed of the other two groups; therefore, in this example, the performance of the model using the third group of parameters is better.

In addition, in order to verify the performance of the model proposed in this article, we compared the method in this article with the DQN algorithm to solve the state transition probability of a stochastic game model. The results are shown in Figure 7.

From Figure 7, we find that the defense strategy value obtained by using the DQN algorithm in the model often performs better than the defense strategy value calculated by using the Double DQN algorithm. That is because the DQN algorithm is likely to cause overestimation. The result is consistent with our expectations. We also found that the model proposed in this paper calculates that the convergence rate of the probability value of the defense strategy is faster than that of the model using the DQN algorithm. Therefore, the above results show that the performance of our proposed model is better.

5. Comparative Analysis of Related Work

The comparison results between the model proposed in this paper and some typical methods are shown in Figure 8 and Table 3.

Figure 8 shows the probability of selecting defense actions in four different methods. From the figure, it can be found that the defense strategy of the method proposed in this article can converge to an equilibrium strategy after about 140 times of learning. The state transition of the method proposed in [8] is a fixed value, which leads to a certain deviation between its result and the objective equilibrium strategy value. In addition, this method assumes complete information, which also makes the calculation result slightly larger than other methods. The method proposed in [9] is an improvement over the method in [8] in certain ways, but the defense strategy obtained by this method is unstable in the early stage and fluctuates greatly. The results are also biased. The result of the method proposed in [28] is better than that of [8, 9] in the early stage of defense. However, as the number of defenses increases, it began to show the disadvantage of this method in processing high-dimensional data. In addition, the Q-learning algorithm is more suitable for dealing with pure strategy problems. When it comes to deal with the mixed strategy

FIGURE 8: Variations in defense strategies of different methods. Take defense action d_2 as an example.

TABLE 2: Bayesian Nash equilibrium of network offense and defense entity.

	a_2	a_3	d_2	d_3
$\sigma_a^*(s_3, \theta_1)$	0.6	0.4	—	—
$\sigma_a^*(s_3, \theta_2)$	0.2	0.8	—	—
$\sigma_d^*(s_3)$	—	—	0.36	0.64

TABLE 3: Comparison between the method in this article and the typical methods.

References	Theory	Game type	Game information	Behavioral rationality	Transition probability	Equilibrium solution	Accuracy	Versatility
[8]	Stochastic game	Static	Complete information	Completely rational	Fixed value	Simple	Poor	Poor
[9]	Stochastic evolutionary game	Dynamic	Incomplete information	Bounded rationality	Dynamic changes	Detailed	Good	Good
[28]	Stochastic game + reinforcement learning	Dynamic	Incomplete information	Bounded rationality	Dynamic changes	Detailed	Good	Good
Our method	Stochastic game + deep reinforcement learning	Dynamic	Incomplete information	Bounded rationality	Dynamic changes	Detailed	Better	Better

TABLE 4: The meanings of the math notations.

Notation	Meaning
s_i	The state of the network system
A_i & D_j	Attacker & defender's action sets
a_i & d_j	The action of attacker & defender
θ_i	Type of attacker
P_A	Prior probability of attacker type
π_a & π_d	The strategy of attacker & defender
π^*	Nash equilibrium strategy
σ_a & σ_d	Probability of attacker & defender choosing action
Q_a & Q_d	The "state-action" revenue function of attacker & defender
V_a & V_d	The "state-strategy" revenue function of attacker & defender
R	Current reward
γ	Discount factor
α	Learning rate

FIGURE 9: Network state transition diagram.

selection problem, the algorithm does not perform well, and the defense strategy would keep oscillating.

As shown in Table 3, we have summarized the four methods. The basic theory of stochastic game used in [8, 9] is relatively simple and the model applicability is general; the basic theory used in [28] is stochastic game and reinforcement learning, and the applicability of the model has been improved. The basic theory used in this paper is to combine stochastic game and deep reinforcement learning. Compared with [8], this article assumes incomplete information as the premise, and the applicability of the model is better. Compared with [9], the model we proposed no longer needs to fix the system state transition probability in advance, which is more in line with the actual situation of network system state transition. Compared with the method proposed in [28], not only is our method in this paper more suitable for processing high-dimensional network system state space, but also the accuracy is higher. In summary, it is more suitable for modeling and analyzing the actual network attack and defense process.

6. Conclusions

This paper analyzes the actual network offensive and defense process modeling requirements and proposes a defense strategy decision model based on incomplete information stochastic game and deep reinforcement learning. This model is aimed at the game problem between the offensive and defensive sides of the network and uses deep reinforcement learning to solve the benefits of the game entities. We have verified that our model is more in line with the modeling requirements of the network offensive and defensive process in the real environment. The model proposed in this paper improves the existing model theoretically. The selected deep reinforcement learning algorithm is more suitable for processing high-latitude game state space, and experiments show that, under the same experimental conditions, the method proposed in this paper has a better convergence rate than other methods in solving the defense equilibrium strategy. Therefore, the research results of this paper provide new research ideas for the selection of network security defense strategies.

In the future, our research will focus on two aspects: (1) to improve the accuracy of the recurring network attack and defense process and (2) to improve the applicability of the model to meet the needs of more complex network environments

Appendix

A. The Meanings of the Math Notations Used in This Paper Are Shown in Table 4

B. Example Calculation

Assume that the attacker type is $\{\theta_1, \theta_2\}$, where θ_1 represents a high-level attacker and θ_2 represents a low-level attacker; the probability distribution of attacker type $(p_a(s_1, \theta_1), p_a(s_1, \theta_2), p_a(s_2, \theta_1), p_a(s_2, \theta_2), p_a(s_3, \theta_1), p_a(s_3, \theta_2), p_a(s_4, \theta_1), p_a(s_4, \theta_2)) = (0.15, 0.85, 0.2, 0.8, 0.3, 0.7, 0.4, 0.6)$. The state set of the network system $S = \{s_1, s_2, s_3, s_4\}$; the network state transition is shown in Figure 9.

The attacker's behavior set $A = \{A_1, A_2, A_3, A_4\}$, where $A_1 = \{a_1, a_2, a_3\}$, $A_2 = \{a_1, a_2\}$, $A_3 = \{a_2, a_3\}$, and $A_4 = \{a_3\}$. The defender's behavior set $D = \{D_1, D_2, D_3, D_4\}$, where $D_1 = \{d_1\}$, $D_2 = \{d_2\}$, $D_3 = \{d_2, d_3\}$, and $D_4 = \{d_1, d_2, d_3\}$.

Next, the authors use the method proposed in this article, taking system state s_3 as an example to solve the Nash equilibrium strategy. Known conditions are as follows:

- (1) The defender's immediate return $R_{a,d}(s_i, a, d, \theta_1)$ when in state s_3 :

$$\begin{aligned}
 R(s_3, \theta_1) &= \begin{bmatrix} R_{a,d}(s_4, a_2, d_2, \theta_1) & R_{a,d}(s_4, a_2, d_3, \theta_1) \\ R_{a,d}(s_4, a_3, d_2, \theta_1) & R_{a,d}(s_4, a_3, d_3, \theta_1) \end{bmatrix} \\
 &= \begin{bmatrix} (20, -22) & (26, -24) \\ (30, -32) & (20, -22) \end{bmatrix}, \\
 R(s_3, \theta_2) &= \begin{bmatrix} R_{a,d}(s_4, a_2, d_2, \theta_2) & R_{a,d}(s_4, a_2, d_3, \theta_2) \\ R_{a,d}(s_4, a_3, d_2, \theta_2) & R_{a,d}(s_4, a_3, d_3, \theta_2) \end{bmatrix} \\
 &= \begin{bmatrix} (10, -14) & (16, -18) \\ (10, -14) & (10, -16) \end{bmatrix}.
 \end{aligned} \tag{B.1}$$

- (2) Considering that in actual network attack and defense events, due to the stochasticity of attack behavior, assume that the attacker's initial strategy in system state s_3 is $(0.5, 0.5)$.
- (3) For parameter settings, see Table 1.
- (4) The transition probability of the system from state s_3 :

$$\begin{aligned}
 p(s_3) &= \begin{bmatrix} p(s_1|s_3, a_2, d_2) & \cdots & p(s_1|s_3, a_3, d_3) \\ \vdots & \ddots & \vdots \\ p(s_4|s_3, a_2, d_2) & \cdots & p(s_4|s_3, a_3, d_3) \end{bmatrix} \\
 &= \begin{bmatrix} 0.4 & 0 & 0.7 & 0 \\ 0 & 0.2 & 0.4 & 0.3 \\ 0 & 0.6 & 0.4 & 0.2 \\ 0.3 & 0.3 & 0 & 0.2 \end{bmatrix}.
 \end{aligned} \quad (B.2)$$

Then, according to the above known conditions (1)–(4), using formula (1) and formula (2), the Bayesian Nash equilibrium in the experimental scene can be obtained, and the results are shown in Table 2.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by National Key Research and Development Program (no. 2018YFC0808306), Hebei Province Key Research and Development Program (19270318D), Hebei Province Internet of Things Monitoring Engineering Technology Research Center (no. 3142018055), and Qinghai Province Internet of Things Key Laboratory (no. 2017-ZJ-Y21) project.

References

- [1] R. Von Solms and J. Van Niekerk, "From information security to cyber security," *Computers & Security*, vol. 38, pp. 97–102, 2013.
- [2] J. Ai, H. Chen, Z. Guo, G. Cheng, and T. Baker, "Mitigating malicious packets attack via vulnerability-aware heterogeneous network devices assignment," *Future Generation Computer Systems*, vol. 111, pp. 841–852, 2020.
- [3] D. E. Denning, "Framework and principles for active cyber defense," *Computers & Security*, vol. 40, pp. 108–113, 2014.
- [4] X. Liang and Y. Xiao, "Game theory for network security," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 472–486, 2013.
- [5] C. T. Do, N. H. Tran, C. Hong et al., "Game theory for cyber security and privacy," *ACM Computing Surveys*, vol. 50, no. 2, pp. 1–37, 2017.
- [6] X. Xu, G. Wang, J. Hu, and Y. Lu, "Study on stochastic differential game model in network attack and defense," *Security and Communication Networks*, vol. 2020, Article ID 3417039, 15 pages, 2020.
- [7] Y.-Z. Wang, C. Lin, X.-Q. Cheng, and B.-X. Fang, "Analysis for network attack-defense based on stochastic game model," *Chinese Journal of Computers*, vol. 33, no. 9, pp. 1748–1762, 2010.
- [8] Y. Fu, Y. Chen, X. Wu, and Y. Song, "Network attack-defense strategies selection based on stochastic game model," *Beijing Youdian Daxue Xuebao/Journal of Beijing University of Posts and Telecommunications*, vol. 37, pp. 35–39, 2014.
- [9] J. Huang and H. Zhang, "A method for selecting defense strategies based on stochastic evolutionary game model," *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, vol. 46, no. 9, pp. 2222–2228, 2018.
- [10] Y. Hu, J. Ma, Y. Guo, and H. Zhang, "Research on active defense based on multi-stage cyber deception game," *Tongxin Xuebao/Journal on Communications*, vol. 41, no. 8, pp. 32–42, 2020.
- [11] L. Wei, A. I. Sarwat, W. Saad, and S. Biswas, "Stochastic games for power grid protection against coordinated cyber-physical attacks," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 684–694, 2018.
- [12] C. Lei, H.-Q. Zhang, L.-M. Wan, L. Liu, and D.-H. Ma, "Incomplete information Markov game theoretic approach to strategy generation for moving target defense," *Computer Communications*, vol. 116, pp. 184–199, 2018.
- [13] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963–1971, 2020.
- [14] C. Luo, Z. Tan, G. Min, J. Gan, W. Shi, and Z. Tian, "A novel web attack detection system for internet of things via ensemble classification," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5810–5818, 2021.
- [15] T. Hua, L. Li, T. Guarda, I. Lopes, and Á. Rocha, "Computer network security technology based on artificial intelligence," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 5, pp. 6021–6028, 2019.
- [16] Y. Zhang and J. Liu, "Optimal decision-making approach for cyber security defense using game theory and intelligent learning," *Security and Communication Networks*, vol. 2019, Article ID 3038586, 16 pages, 2019.
- [17] A. G. Perevozchikov, V. Y. Reshetov, and I. E. Yanochkin, "Multilayered attack-defense model on networks," *Computational Mathematics and Mathematical Physics*, vol. 59, no. 8, pp. 1389–1397, 2019.
- [18] P. P. Chakraborty and K. Roy, "Discretization based solutions for secure machine learning against adversarial attacks," *IEEE Access*, vol. 7, pp. 70157–70168, 2019.
- [19] S. Lall and A. Goldsmith, "Networked markov decision processes with delays," *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 1013–1018, 2012.
- [20] J. Clifton and E. Laber, "Q-learning: theory and applications," *Annual Review of Statistics and Its Application*, vol. 7, no. 1, pp. 279–301, 2020.
- [21] L. Zhang, L. Tang, S. Zhang, Z. Wang, X. Shen, and Z. Zhang, "A self-adaptive reinforcement-exploration Q-learning algorithm," *Symmetry*, vol. 13, no. 6, p. 1057, 2021.
- [22] S. Yoon, J.-H. Cho, D. S. Kim, T. J. Moore, F. Free-Nelson, and H. Lim, "Attack graph-based moving target defense in software-defined networks," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1653–1668, 2020.
- [23] A. Vasudeva and M. Sood, "Survey on sybil attack defense mechanisms in wireless ad hoc networks," *Journal of Network and Computer Applications*, vol. 120, pp. 78–118, 2018.
- [24] D. Chasaki and T. Wolf, "Attacks and defenses in the data plane of networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 798–810, 2012.
- [25] Y. Sun, W. Xiong, Z. Yao, K. Moniz, and A. Zahir, "Network defense strategy selection with reinforcement learning and pareto optimization," *Applied Sciences*, vol. 7, no. 11, p. 1138, 2017.

- [26] J. Zhang, C. Zhang, and W. Chien, "Overview of deep reinforcement learning improvements and applications," *Journal of Internet Technology*, vol. 22, no. 2, pp. 239–255, 2021.
- [27] X. Wang, Y. Gu, Y. Cheng, A. Liu, and C. L. P. Chen, "Approximate policy-based accelerated deep reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 1820–1830, 2020.
- [28] H. Zhang, J. Yang, and C. Zhang, "Defense decision-making method based on incomplete information stochastic game and Q-learning," *Tongxin Xuebao/Journal on Communications*, vol. 39, no. 8, pp. 56–68, 2018.