Hindawi
Security and Communication Networks
Volume 2022, Article ID 1886516, 13 pages
https://doi.org/10.1155/2022/1886516

Research Article

DDoS Defense Method in Software-Defined Space-Air-Ground Network from Dynamic Bayesian Game Perspective

Zhaobin Li,[1,2] **Bin Yang**,[1] **Xinyu Zhang**,[1] **and Chao Guo**[1,2]

[1]*Communication Engineering Department, Beijing Electronics Science and Technology Institute, Beijing 100070, China*
[2]*The State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710126, China*

Correspondence should be addressed to Chao Guo; guo99chao@163.com

Received 29 October 2021; Accepted 15 December 2021; Published 7 January 2022

Academic Editor: Zhen Wang

The centralized management of Software-Defined Network (SDN) brings convenience to Space-Air-Ground Integrated Networks (SAGIN), which also makes it vulnerable to Distributed Denial of Service (DDoS). At present, the popular detection methods are based on machine learning, but most of them are fixed detection strategies with high overhead and real-time control, so the efficiency is not high. This paper designs different defense methods for different DDoS attacks and constructs a multitype DDoS defense model based on a dynamic Bayesian game in the Software-Defined Space-Air-Ground Integrated Networks (SD-SAGIN). The proposed game model's Nash equilibrium is solved based on the different costs and payoffs of each method. We simulated the attack and defense of DDoS in Ryu controller and Mininet. The results show that, under our model, the attacker and defender's strategies are in a dynamic balance, and the controller can effectively reduce the defense cost while ensuring detection accuracy. Compared with the existing traditional Support Vector Machine (SVM) defense method, the performance of the proposed method is better, and it provides one of the references for DDoS defense in SD-SAGIN.

1. Introduction

Software-Defined Space-Air-Ground Integrated Networks (SD-SAGIN) employ Software-Defined Network (SDN) in Space-Air-Ground Integrated Networks (SAGIN) to achieve highly centralized management. SAGIN [1] combines the advantages of three networks with different heights. It has a wide coverage and few transmission restrictions, but it also has disadvantages such as high delay, instability, and small throughput.

At the same time, the complexity and scale of the space-ground network also bring great challenges to the management of the Network; thus, the Software-Defined Network (SDN) has been introduced as a solution [2]. Different from the closure and complexity of the traditional network, SDN separates the data plane from the control plane. It has the advantages of high openness, easy management, and programmability, which enhance the research and development of the network and reduce the cost of maintenance and update. It can provide further service innovation and business flexibility for the integrated network of SAG. But high concentration also brings serious security issues. The controller is the brain of the whole SD-SAGIN. Once the controller is attacked or even destroyed, the whole network will suffer.

One of the most serious threats to controllers is Distributed Denial of Service (DDoS). It is caused by a large number of controlled puppets hosts to attack the targeted system, resulting in a sharp consumption of the targeted system's resources, and finally make its service quality decline or even terminated. Due to the centrality of SDN, when a DDoS attack occurs, the switches will generate a large number of packet-in messages to send to the controller, resulting in the controller's resources being occupied, the flow table of the switch increases sharply, and a large number of messages will block the secure channel between the controller and switches, which may eventually paralyze the whole SDN. If DDoS occurs in SD-SAGIN, communication channels may be blocked in a short time, and controllers' resources would be occupied, causing service quality to be severely damaged.

As a mathematical tool, game theory is an analytical tool that formulates the interaction between various incentive structures with the idea of mathematical modeling to solve or understand the optimization strategies of each party. In recent years, it has been more and more introduced into the field of computer science and plays an important role in network attack and defense, resource allocation, and so on. Dynamic Bayesian game is a type of game in which a player determines his choice of strategy by estimating the probabilities (prior probabilities) of other players' types, predicting the strategies chosen by other players, and calculating the maximum average payoffs under his strategy. In this paper, a dynamic Bayesian game of DDoS attack and defense in SD-SAGIN is constructed. Compared with the existing traditional Support Vector Machine (SVM) defense method, the proposed method can reduce defense overhead and improve defense efficiency while ensuring network security. The traditional SVM is used for proving that, with the help of game theory, the defense of DDoS based on machine learning could be more economically.

In this paper, in order to reflect the effect of the method, we simplified the implementation process of the model. It is assumed that "attack is defended as soon as detected" and the defense method is reflected in the form of detection. When the controller detects an attack, it will stop the attack, and the stop means are not described in this paper, such as tracing the source and closing the attacker's port.

The rest of this paper is organized as follows: Section 2 introduces some related works. In Section 3, the offense and defense game model is described. The simulation and analysis are given in Section 4. At last, a conclusion is drawn in Section 5.

2. Related Work

At present, the mainstream methods used for DDoS detection in SDN are divided into two categories: the detection methods based on statistical information and the detection methods based on machine learning [3]. The detection methods based on statistical information can well reflect the randomness of the sequenced queues through information entropy.

The studies in [4, 5], respectively, proposed methods for detecting DDoS attacks in SDN based on information entropy. These methods mainly calculate the entropy of the collected data. When the entropy is less than a certain threshold, it indicates that the randomness of traffic is small, and it can be judged that a DDoS attack has occurred.

Compared with the detection methods based on information entropy, the detection methods based on machine learning are used more frequently. Yuhua combined KNN and FKNN algorithms to deploy DDoS detection and mitigation mechanisms in the application layer and control layer with high efficiency and stability. However, KNN takes a large amount of computation and high cost [6]. The authors of [7, 8] put forward their DDoS attack detection methods based on a decision tree algorithm. These methods used the mapping relationship between object attributes and object values in the decision tree algorithm to extract information from flow table entries for classification to detect DDoS traffic. The authors of [9–11] used deep learning algorithms such as neural networks to solve the problem of DDoS detection in SDN. This kind of method has high accuracy, which is better than the traditional machine learning method. At the same time, it can also shorten the processing time of classification detection, but the number of features is too big, which increases the detection cost and is not efficient enough. The study in [12] proposed a DDoS attack situation assessment method based on the optimized cloud model of the impact function, and the V-Support Vector Machines (V-SVM) classification model was established. Three evaluation indicators are proposed, and an index weight calculation algorithm is used to measure the importance of different indicators. This method can not only improve the detection rate and false-negative rate of DDoS attacks but also effectively deal with DDoS. The attack situation is more accurate and more flexible than existing methods.

In recent years, game theory, as an optimal strategy selection method, has been gradually introduced into the DDoS defense of SDN. Marcos et al. [13] proposed a DDoS attack mitigation mechanism by using game theory. Defenders would choose the best defense strategy, but the game theory was not used in the anomaly detection part. The authors of [14, 15] put forward an anomaly detection mechanism and a pseudohoney pot trapping system, respectively, aiming at the DDoS attack problem faced by the Internet of Things based on SDN, and the game theory was used as the optimal strategy selection method. Ankur et al. [16] used the programmability of SDN to propose a game theory model for attack analysis and countermeasure selection. The model is based on the rewards and punishments of the multiparty dynamic game, and it can effectively meet the expectations of network administrators by using anonymous theorem to punish DDoS traffic, reduce the bandwidth of attackers, and reward cooperators. The study in [17] proposed a DDoS defense mechanism in SDN smart grids. It establishes a noncooperative dynamic Bayesian game model between a DDoS attacker and a network hypervisor, so that the defender can make use of the limited resources efficiently.

In the satellite network, the study in [18] generated an efficient and energy-saving SDSN topology based on the DCTG algorithm and the characteristic formula of the optimization target. An improved network topology generation algorithm was proposed. Meanwhile, based on the deep reinforcement learning (DRL) algorithm of the DDoS mitigation strategy, a DDoS mitigation mechanism for satellite networks was proposed, which could effectively alleviate the abnormal traffic caused by DDoS attacks in SDSN and reduce the extra energy consumption of satellite nodes in handling abnormal traffic. The authors of [19] proposed a satellite network topology optimization algorithm combined with NIDS based on federated learning distributed NIDS in STN. This algorithm can reasonably allocate resources in each domain and analyze and block malicious traffic. It can also reduce the difficulty of malicious packet tracking caused by frequent link switching.

Compared with traditional NIDS, it can achieve higher accuracy of malicious traffic identification, but lower CPU utilization. To evaluate the trusted path, based on the traditional QoS model, the study in [20] proposed a TRM-based trusted routing (TR) model and a hybrid routing (HR) model based on TR and QoS model, which can protect the normal traffic in ISTN from attack. The authors in [21] proposed a software-defined Space-Air-Ground Integrated network architecture, which provided data relays for missions such as deep space exploration, ensuring the anti-sabotage capability of stratospheric communication, and established a large-scale information network with mutual assistance and complementary functions.

To sum up, there are many DDoS attack detection technologies in the SDN and SAGIN environment, but most of them have problems such as long detection time, low accuracy, and high overhead. As we know, for a network, it does not suffer from DDoS attacks all the time. But to protect itself from types of DDoS attacks, it has to put the defense system in a working state consuming the highest cost, just for defending some possibilities. No doubt this has caused some unnecessary waste of system resources. Therefore, if there is a kind of defense system including a variety of DDoS attack detection methods, we can employ these methods differentially and do not need to maximize the cost to maintain the defense system all the time. In this way, at the expense of less attack detection time, the cost of defenders will be reduced more, and finally, the whole optimization of the defense system will be achieved.

In this paper, a DDoS attack defense method based on a dynamic Bayesian game is proposed. Attackers and defenders are regarded as two sides of the game, and corresponding detection methods are set for different types of DDoS attacks. These detection methods are specific and cannot detect each other. The mathematical model of offense and defense game between the two sides is established, and the attack and defense strategies are scheduled on this basis. The controller takes the role of global supervisory in SDN, and the defense cost is reduced, while the detection accuracy is guaranteed.

3. Design of Attack and Defense Model

In recent years, game theory has been introduced into the field of network security more and more because of its characteristics in selecting optimal strategies. By modeling, analyzing, and solving the strategies, actions, and payoffs of the game players, game theory guides all parties to seek to maximize their own payoffs and finally reach the equilibrium state.

In this paper, a DDoS attack and defense model based on a dynamic Bayesian game in SD-SAGIN is proposed. Three typical DDoS attack modes are selected, and three detection methods based on the SVM algorithm are designed for them. Under the guidance of game theory, both sides select these offensive and defensive methods and tend to maximize their own payoffs, thus forming Nash equilibrium. Under the Nash equilibrium, the strategy selection of both sides presents a characteristic of dynamic equilibrium; that is, when the defender's prior belief (about whether there is an attacker) is high, the attacker chooses not to attack with high probability and chooses to attack under the low prior belief, thus reducing the monitoring overhead of the defender and improving the defense efficiency.

In the SD-SAGIN DDoS game model, the two sides of the game are the attacker and the defender, the attacker is the initiator of DDoS, and its action set is {Not Attack, SYN Flood, ICMP Flood, UDP Flood}. The defender is the SDN controller, whose set of actions is {NOT DETECT, SYN DETECT, ICMP DETECT, UDP DETECT}. Each detection method can only detect the corresponding attack method. For example, SYN Detect can only detect SYN Flood, while ICMP Detect and UDP Detect will consider SYN Flood as normal traffic. Support Vector Machine (SVM) is used as the detection method. The type of defender is certain and must be in the game all the time, but the type of attacker is uncertain. The attacker can choose to participate in the game or not participate in the game. This is an incomplete information game, so the Bayesian game is chosen to model this problem.

The goal of the defender is to predict as accurately as possible what the attacker will do in the next stage based on the current situation and deploy the corresponding defensive action. On the other hand, the target of the attacker is to predict the defense strategy and adjust the attack strategy to consume the controller's network resources as much as possible while minimizing its own cost. The attacker may choose to reduce the intensity of the attack in order not to expose himself, while the defender is detecting. Thus, a set of Bayesian dynamic games is formed. In this game, the controller as the defender (D) is not clear whether he is gaming with the attacker (A); it will be according to the observed network situation, which is the revision of the attacker's belief about the existence of the attacker. σ is the prior belief, and $\sigma\prime$ is a prior belief revision. Table 1 shows the symbolic representations in this game. Figure 1 shows the overall design of the system.

3.1. Elements of the Game. A complete game should consist of players, strategies, and payoffs:

(1) Players: players consist of DDoS attacker (A) and SD-SGAIN controller (D). They rationally decide their next strategy based on the actions of their opponents, with the goal of maximizing their own gains.

(2) Strategies: the strategy set of attacker A is $\{a_0, a_1, a_2, \ldots, a_N\}$, where a_m $(1 \le m \le N)$ represents using a DDoS attack-method-m. a_0 represents not attacking. $\{p_0, p_1, p_2, \ldots, p_N\}$ represents the probability distribution of the attacker's actions. p_m $(1 \le m \le N)$ represents the probability of taking the attack-method-m, p_0 represents the probability of not attacking, and

$$p_0 + \sum_{1}^{N} p_m = 1. \tag{1}$$

Table 1: Notations and significance.

Notations	Significance
A	DDoS attacker
D	Defender (SAG-SDN controller)
a_m	Using attack-method-m
d_n	Using defense-method-n
p_m	Probability of using attack-method-m
q_n	Probability of using defense-method-n
c_a^m	Cost of using attack-method-m for attacker
c_d^n	Cost of using defense-method-n for defender
w^m	Utility of attack-method-m's success
P	Penalty for attacker after being detected
R	Total resources of SAG-SDN controller
N	Numbers of attack methods
U_A	Utility of DDoS attacker
U_D	Utility of defender
σ	Prior belief of defender about the type of network
$\sigma\prime$	Correction of prior belief

The strategy set of defender D is $\{d_0, d_1, d_2, \ldots, d_N\}$, where d_n $(1 \le n \le N)$ represents the deployment of a DDoS detection-method-n and d_0 represents not defending. $\{q_0, q_1, q_2, \ldots, q_N\}$ represents the probability distribution of the defender's actions, q_n $(1 \le n \le N)$ represents the probability of taking detection-method-n, q_0 represents the probability of not defending, and

$$q_0 + \sum_{1}^{N} q_n = 1. \tag{2}$$

(3) Payoffs: the payoffs of the players depend on the strategies they choose. First of all, we make it clear that when the attacker takes a_m $(1 \le m \le N)$ and the defender takes d_n $(1 \le m \le N)$, the defender succeeds, and the attacker fails only when $m = n$, which means the defender taking the detection method corresponding to the attack method. When m≠n, the defender fails and the attacker succeeds. When the attacker takes a_m $(1 \le m \le N)$ successfully, the gain is w^m and the cost is c_a^m. And when the defender takes d_n $(1 \le n \le N)$ successfully, the revenue is w^m, and the cost is c_d^n. When the defender detects an attack, he will impose additional punishment P on the attacker. In this model, we assume that once the attack is successful, the resource R of the whole controller will be unavailable, so there is $w^m = R$. We will discuss it in the following two cases.

Scenario 1. An attacker exists, and the defender's prior belief is σ. When $m = n$, the defender succeeds, and the attacker's gain equals the cost of executing the attack-method-m plus the penalty P imposed by the defender. The defender's gain equals the controller resource R minus the cost of deploying defense-method-n. If $m \neq n$, the attacker succeeds, and the attacker's gain equals the controller resource R minus the cost of executing the attack-method-m. The defender loses the cost of the controller resource R and its deployment defense-method-n.

Scenario 2. An attacker does not exist, in which case the defender has a belief of $1 - \sigma$. There is no attacker in the game, the attacker's gain is always 0, and the defender's gain is 0 (when not defending), or 0 minus cost of detection-method-n (when defending). In both cases, the attacker-defender payoff matrix is shown in Table 2.

3.2. Payoff Functions. In the game model, it is very important to construct the payoff functions according to the game elements, and the actions of both sides are dominated by the payoff functions.

3.2.1. Attacker's Payoff Function. When the attacker selects method a_0 (not attack), the attacker's payoff is

$$U_A(a_0) = \sigma * 0 + (1 - \sigma) * 0 = 0. \tag{3}$$

When the attacker selects method a_m $(1 \le m \le N)$, the attacker's payoff is

$$U_A(a_m) = \sigma\left[U(a_m, d_{n=m}) + \sum_{n=1}^{n=N} U(a_m, d_{n\neq m})\right] + (1 - \sigma) * 0. \tag{4}$$

Substituting the matrix payoff in Table 2, the attacker's payoff is

$$U_A(a_m) = \sigma\left[q_{n=m}\left(-P - c_a^m\right) + \left(1 - q_{n=m}\right)\left(R - c_a^m\right)\right]. \tag{5}$$

Then, the total payoff of the attacker is

$$\begin{aligned} U_A &= \sum_{m=0}^{N} U_A(a_m) \\ &= \sigma \sum_{m=1}^{N} \left[-q_{n=m}(P + R) + \left(R - c_a^m\right)\right]. \end{aligned} \tag{6}$$

3.2.2. Defender's Payoff Function. When the defender chooses method d_0 (not defend), the defender's payoff is

$$\begin{aligned} U_D(d_0) = \sigma\left\{\sum_{m=1}^{N}\left[U_D(d_0, a_m)\right] + U_D(d_0, a_0)\right\} \\ + (1 - \sigma)\left[U_D(d_0, a_0)\right]. \end{aligned} \tag{7}$$

When the defender selects method d_n $(1 \le n \le N)$, the defender's payoff is

$$\begin{aligned} U_D(d_n) = \sigma\left[\sum_{m=1}^{N} U_D(d_n, a_m) + U_D(d_n, a_0)\right] \\ + (1 - \sigma)\left[U_D(d_n, a_0)\right]. \end{aligned} \tag{8}$$

Substituting the matrix payoff in Table 2, the defender's payoff is

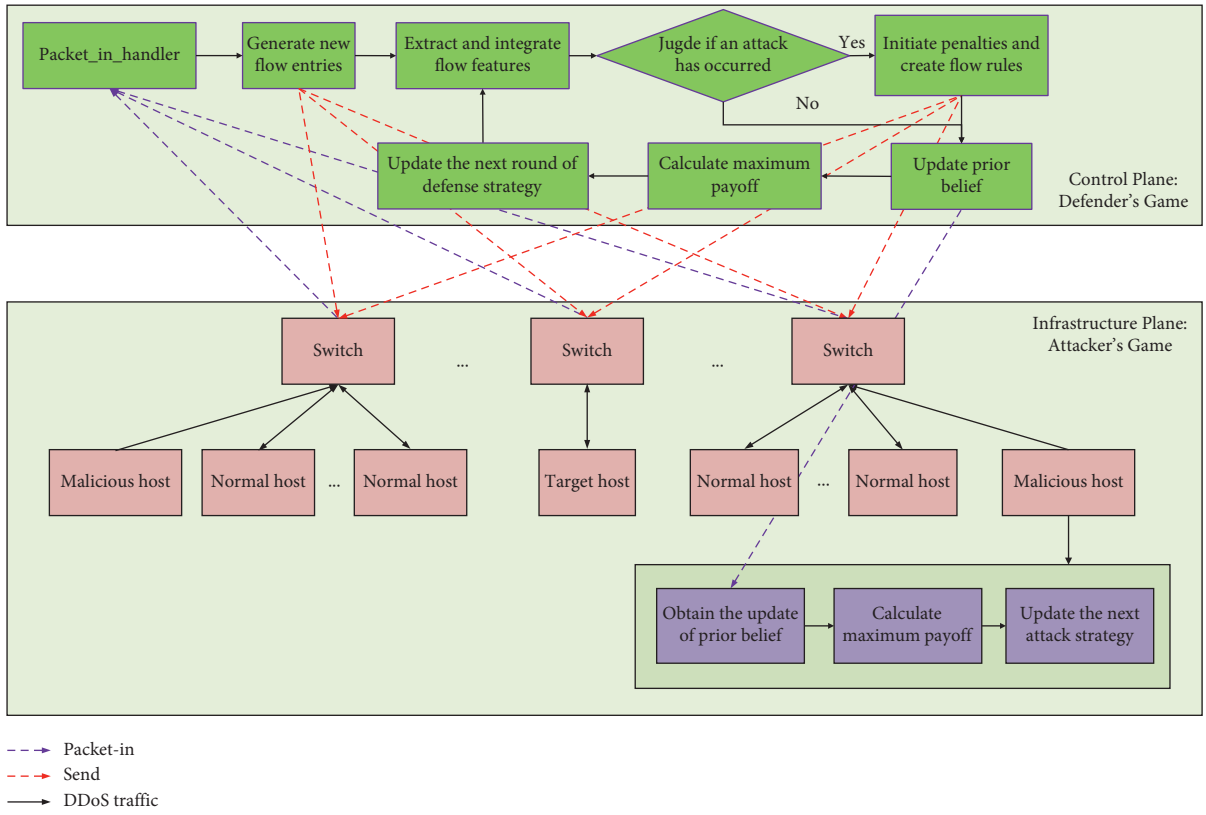

Figure 1: Framework of the system.

Table 2: Attacker-defender payoff matrix.

Attacker/defender	Detect	Not detect
Attack	$-P-c_a^m, R-c_d^n\ (m=n)$ $R-c_a^m, -R-c_d^n\ (\mathrm{m}\neq\mathrm{n})$	$R-c_a^m, -R$
Not attack	$0, -c_d^n$	0, 0

$$U_D(d_n)=\sigma\left\{p_{n=m}\left(R-c_d^n\right)+\sum_{m\neq n}p_m\left(-R-c_d^n\right)+p_0\left(-c_d^n\right)\right\}+(1-\sigma)\left(-c_d^n\right). \tag{9}$$

Then, the total payoff of the defender is

$$\begin{aligned}U_D&=\sum_{n=1}^{N}U_D(d_n)+U_D(d_0)\\&=\sum_{n=1}^{N}\sigma p_{n=m}R-\sum_{n=1}^{N}c_d^n.\end{aligned} \tag{10}$$

3.3. Nash Equilibrium

Theorem 1. *Nash equilibrium refers to a group of strategy combinations in which all players choose the strategy that maximizes their returns and have no reason to deviate from this strategy.*

Because the existence of the attacker is uncertain, there is no pure Nash equilibrium strategy in the game model but only a mixed Nash equilibrium strategy. Mixed strategy refers to the strategy in which players randomly choose different actions in a certain probability distribution under the given information, and it is a probability distribution in its strategy space.

3.3.1. Defender's Mixed Strategy. According to the payoff function of the attacker, the mixed strategy of the defender $\{q_0, q_1, q_2, \ldots, q_N\}$ is to be solved. For the attacker, no matter what method the defender chooses, it can obtain the maximum expected payoff; that is, the payoff of each attacker's action is equal.

To let $U_A(a_N)=\cdots=U_A(a_2)=U_A(a_1)=U_A(a_0)$,

$$U_A(a_m)=\sigma\left[q_{n=m}\left(-P-c_a^m\right)+\left(1-q_{n=m}\right)\left(R-c_a^m\right)\right]=0. \tag{11}$$

Solving (11),

$$q_{n=m} = \frac{R - c_a^m}{P + R}\ (1 \le m \le N),$$
$$q_0 = 1 - \sum_{m=1}^{N} q_m. \tag{12}$$

It can be seen from formula (12) that the probability of the defender choosing a certain defense method is related to the cost of the corresponding attack method and the punishment imposed by the defender. The higher the attack cost is and the greater the punishment is, the less likely the attacker is to choose this method and the less likely the defender is to defend this attack.

3.3.2. Attacker's Mixed Strategy. According to the defender's payoff function, the attacker's mixed strategy $\{p_0, p_1, p_2, \ldots, p_N\}$ is to be solved. For the defender, no matter what method the attacker chooses, it can obtain the maximum expected payoff; that is, the payoff of each action of the defender is equal.

To let $U_D(d_N) = \cdots = U_D(d_2) = U_D(d_1) = U_D(d_0)$,

$$\sigma\left\{p_{n=m}\left(R - c_d^n\right) + \sum_{m\neq n} p_m\left(-R - c_d^n\right) + \left(1 - \sum_{m=1}^{N} m = 1\right)\left(-c_d^n\right)\right\} + (1-\sigma)\left(-c_d^n\right) = \sigma\left\{\sum_{m=1}^{N}\left[p_m(-R)\right]\right\}. \tag{13}$$

Solving (13),

$$p_{n=m} = \frac{c_d^n}{2\sigma R}\ (1 \le n \le N),$$
$$p_0 = 1 - \sum_{m=1}^{N} p_m. \tag{14}$$

It can be seen from formula (14) that the probability of an attacker choosing an attack method is related to the cost of the corresponding defense method and the defender's belief σ about the existence of an attacker. The higher the defense cost is and the lower the belief is, the less likely the defender is to choose the defense method and the more likely the attacker is to deploy the attack.

To sum up, the mixed Nash equilibrium strategy set of both offensive and defensive sides is $\{[p_0, p_{m=n} = c_d^n/2\sigma R\,(1 \le n \le N)]; \quad [q_0, q_{n=m} = R \quad -c_a^m/P + R\,(1 \quad \le m \le N)]\}$. In other words, the two players will randomly choose their actions with different probabilities at a certain stage of the game.

3.4. Prior Belief Modification. The most important thing in the dynamic Bayesian game is the correction of the prior belief σ'.

Before the game of the next stage starts, the defender modifies the belief of the existence of the attacker according to the observed actions of the attacker currently and accordingly changes his defensive strategy of the next stage.

Let the network state $S_A = \{0, 1\}$, where $S_A = \{0\}$ represents that there is no attacker and $S_A = \{1\}$ represents that there is an attacker. While the controller (defender) is always normal, $S_D = \{1\}$. The detection rate of the defense system is P_D, and the false alarm rate is P_F. The defender will judge whether the attacker participates in the game according to the detected network state.

The historical action set of attacker A at the time t_t is

$$h_A(t_t) = \{a_A(t_0), a_A(t_1), \ldots, a_A(t_{t-1})\}, \tag{15}$$

where $a_A(t_t)$ represents the network state detected by the defender in t_t; $a_A(t_t) = \{0, 1\}$, $\{0\}$ represents there is no attack and $\{1\}$ represents there is an attack. The probability distribution of the attacker's optional action in t_n under different states is as follows:

$$\begin{aligned}
&P\left(a_A(t_t) = 1 | S_A = 1\right) = P_D * p_0 + P_F * (1 - p_0),\\
&P\left(a_A(t_t) = 0 | S_A = 1\right) = (1 - P_D)p_0 + (1 - P_F)(1 - p_0),\\
&P\left(a_A(t_t) = 1 | S_A = 0\right) = P_F,\\
&P\left(a_A(t_t) = 0 | S_A = 0\right) = 1 - P_F,
\end{aligned} \tag{16}$$

where $P(a_A(t_t)|S_A)$ represents the probability of the detected network state $a_A(t_t)$ when the real network state is S_A. The prior belief of t_{t+1} is the prior belief modification of t_n. According to the Bayesian formula, the prior belief of the defender in t_t on whether there is an attacker is modified as

$$\sigma' = \sigma_D\left(S_A | a_A(t_t), h_A(t_t)\right) = \frac{\sigma_D\left(S_A | h_A(t_t)\right) P\left(a_A(t_t) | T_A, h_A(t_t)\right)}{\sum \sigma_D\left(S_A | h_A(t_t)\right) P\left(a_A(t_t) | T_A, h_A(t_t)\right)}. \tag{17}$$

This is the prior belief of the defender in t_{t+1}, which will guide the strategic choice of the defender in the next stage. The algorithm pseudocode of optimal strategy selection for both attacker and defender in t_{t+1} is as follows.

Firstly, input initial parameters: defender's prior belief σ, costs of an attack c_a, defense costs c_d, punishment P, controller's resource R, initial attack strategy space a and defense strategy space d, detection rate P_D, and false alarm rate P_F. Secondly, both sides calculate their biggest payoffs of the current stage according to the current prior belief, then output strategy spaces A^* and D^* of the next stage, and

Input: σ, c_a, c_d, P, R, P_D, P_F
Output: A^*, D^* //Optimal attack and defense strategies
begin
(1) $a = \{a_0, a_1, a_2, \ldots, a_N\}$//Attack type space
(2) $d = \{d_0, d_1, d_2, \ldots, d_N\}$//Defense type space
(3) Compute $\max U_A$, $\max U_D$ //attacker and defender's maximum benefits
(4) Output $p = \{p_0, p_1, p_2, \ldots, p_N\}$//The attacker's current round optimal mixed strategy set
(5) Output $q = \{q_0, q_1, q_2, \ldots, q_N\}$//The defender's current round optimal mixed strategy set
(6) Deploy A^*, D^*
(7) Defender detection results $a_A(t_t) = \{0, 1\}$ //Check if an attack occurred. 0 did not occur. 1 did occur
(8) **if** $a_A(t_t) == 1$
(9) $\sigma' = \text{Bayesian}(S_A | a_A(t_t) = 1)$ //Bayes' rule is used to calculate the prior probability of the presence of attackers
(10) **else**
(11) $\sigma' = \text{Bayesian}(S_A | a_A(t_t) = 0)$
(12) **end if**
(13) $\sigma = \sigma'$ //A transcendental belief correction, that is, the next round of defender's transcendental belief
end

Algorithm 1: The optimal strategy selection algorithm for both sides in t_t

randomly select actions of offense and defense. Thirdly, according to the detected network situation (0 means no attack is detected and 1 means attack is detected), the defender updates the prior belief by using the Bayesian formula, and then both sides play the next round of the game in cycles (see Algorithm 1).

3.5. DDoS Attack Detection. Different types of DDoS attacks have different flow characteristics, which are not differentiated in most studies at present. The flows of DDoS are regarded as a whole, and the overall defense method is adopted. It requires the collection of the most features of samples to implement real-time monitoring, which costs a lot. In this paper, for different types of DDoS attacks, the traffic characteristics that best describe the attack mode are extracted, respectively, for detection cost differentiation, so as to provide support for the selection of different strategies in the game theory model. Three common DDoS attack methods are selected, namely, SYN Flood, ICMP Flood, and UDP Flood. Their traffic characteristics are shown in Table 3. SYN Flood has three features, ICMP Flood has four, and UDP Flood has five. The main attack features selected are as follows. The samples required for the calculation of the following features are collected by the controller within a time interval.

Average Packets per Flow (AP). When a DDoS attack occurs, the number of flows increases sharply, resulting in a decrease in the average packet number of flows.

$$\text{AP} = \frac{n}{f}, \tag{18}$$

where n represents the number of packets and f represents the number of flows.

Average Bytes of Packet per Flow (AB). When a DDoS attack occurs, the number of flows increases sharply, and the information load in the attack traffic is very small, which will lead to the decrease of the average bit number of stream packets.

$$\text{AB} = \frac{b}{f}, \tag{19}$$

where b represents the number of packet bits.

Flow Entry Growth (FG). When the network situation is normal, the growth rate of flow tables is low and relatively stable. When a DDoS attack occurs, the number of flow entries will increase sharply, which will increase the burden on the controller and affect the quality of the network.

$$\text{FG} = \frac{e}{t}, \tag{20}$$

where e represents the number of additional flow entries and t represents interval time.

Source IP Addresses Growth (SG). Some DDoS will use a large number of forged source IP addresses to attack the target host, resulting in a sharp increase in the growth rate of source IP addresses in the flow table.

$$\text{SG} = \frac{s}{t}, \tag{21}$$

where s represents the number of additional source IP addresses.

Destination IP Addresses Growth (DG). DDoS generally aims at a certain host or several hosts. During the attack period, the destination IP addresses will present a centralized situation, and the growth rate will slow down.

$$\text{DG} = \frac{d}{t}, \tag{22}$$

where d represents the number of additional destination IP addresses.

We take 2 seconds as an interval to extract and integrate the above features, respectively. The sample-set calculated by collection is $S = \{(X_i, Y_j), i = 1, 2, \ldots, k, j = 1, 2\}$, where X_i is the feature tuple, SYN Flood's tuple is [FG, SG, DG], ICMP Flood's tuple is [AP, AB, FG, SG, DG], and UDP Flood's tuple is [AP, AB, FG, SG, DG]; Y_j is the classification

identification, $Y_j = 0$ represents the classification of normal traffic and $Y_j = 1$ represents the classification of attack traffic. SVM is selected for training, which is a small sample learning method. SVM can avoid "dimension disaster" and has a good performance in dichotomy problems. To ensure the training efficiency, at least 4000 records are collected for each type of traffic, and the ratio of the number of training records to the testing records is 2 : 1. The classification results of the three DDoS attacks are shown in Table 4. It can be seen that the accuracy is high. The three detection methods are independent of each other and cannot be tested by each other.

4. Experiment and Analysis

This simulation is completed with a virtual machine on VMware on a desktop, and the configurations of the desktop and the virtual machine are shown in Table 5 and 6.

In this paper, the network simulation tool is Mininet, the controller is local Ryu, and the DDoS attack generation tool is hping3. A fan-shaped topology is set up, which consists of a controller C0, three Open vSwitch switches S1, S2, and S3, and 15 hosts h1–h15. Since the controller is local, its IP address is 127.0.0.1:6653, and the IP addresses of the three switches are 121.0.0.11, 122.0.0.11, and 123.0.0.11. Switch S1 is connected to five hosts from h1 to h5 with IP addresses from 121.0.0.1 to 121.0.0.5. Switch S2 is connected to five hosts from h6 to h10 with IP addresses from 122.0.0.1 to 122.0.0.5. Switch S3 is connected to five hosts from h11 to h15, and IP addresses are 123.0.0.1~123.0.0.5. S1 is connected with S2 and S3, respectively. The targeted host h1's IP address in the S1 network is 121.0.0.1. Attack hosts are h6, h9, and h11. The bandwidth between hosts in the same network segment is about 25 Gbit/s, and the bandwidth between hosts in different network segments is about 100 Mbit/s. The test network topology is shown in Figure 2.

CPU usage can be a good indicator of the resources consumed by a process. Thus, the different costs in this paper are set as the CPU usage of the corresponding processes. According to formula (12), the probability of the defender choosing a certain defense method depends on the cost of the corresponding attack method, the total amount of controller resources, and the punishment cost of the attacker. The cost of the attack method refers to the utilization rate of the hping3 tool in the CPU when an attack is launched alone. The total resource of the controller is set as 1 (when the whole resource of our computer is occupied by the Ryu-manager process). The cost of the attacker being punished is mainly that the channel of message transmission between the two is closed by the controller. In our topology, there are three malicious hosts, so set $P = 2.5$, a little smaller than 3 for space reservation. According to actual statistical calculation, when hping3 is used to initiate SYN Flood, ICMP Flood, and UDP Flood for one minute, the average CPU occupancy is 1.573%,1.483%, and 1.558%. Thus, the defense strategy can be calculated as follows: $\{q_0 = 0.1560,\ q_{\mathrm{SYN}} = 0.2812,\ q_{\mathrm{ICMP}} = 0.2815,\ q_{\mathrm{UDP}} = 0.2813\}$.

According to formula (14), the probability of an attacker taking an attack method depends on the cost of the defense method, the number of controller resources, and the defender's prior belief. The initial prior belief is set as 0.5 (intermediate probability), and the controller resource is 1. According to the actual statistical calculation, the controller uses SYN Detect, ICMP Detect, and UDP Detect for one minute, and their average CPU occupancy rates are 7.208%, 7.373%, and 7.398%. Then, the initial attack strategy can be calculated as $\{p_0 = 0.7820,\ p_{\mathrm{SYN}} = 0.0721,\ p_{\mathrm{ICMP}} = 0.0737,\ p_{\mathrm{UDP}} = 0.0740\}$. The prior belief update is affected by the detection rate and false alarm rate, and the detection rate is set as $P_D = 0.9$, which means that the probability of detecting at the current stage is 0.9. The false alarm rate is almost 0 from Section 3.5, which can be set as $P_F = 0.001$. So far, all the initial parameters required for the model have been input.

4.1. DDoS Attack Detection. The prior belief correction of the defender for the presence of an attacker in the network is determined by formula (17), and the simulation results are shown in Figure 3.

In Figure 3, the blue ○ line represents the defender's prior belief about the presence of an attacker in the network, which can be seen as changing with the observed network state. Initially, the defender does not detect the attack, so its belief in the presence of an attacker in the network drops and stays low. Once an attack is detected, the prior belief rapidly rises to 1, and the defender holds a belief that there must be an attacker in the network and then maintains full defense for a longer period. After failing to detect the attack for a long time, the prior belief gradually decreases and remains at a low level and then rises rapidly after the attack occurs again. It can be seen that the prior belief correction has high sensitivity and fast convergence speed and can effectively deal with the change of attack situation, thus saving the unnecessary waste of resources. The red + line represents the probability that the attacker does not attack; black · line, green ∗ line, and yellow < line, respectively, represent the probability that the attacker takes SYN Flood, ICMP Flood, and UDP Flood. It can be seen from formula (14) that the probability of an attacker choosing a certain attack method is related to the defender's prior belief and the cost of the corresponding defense method. In the case of a small difference in defense cost, the probability curves of the three attack methods are similar. As can be seen from Figure 3, with the change of the defender's prior beliefs, the attacker will try to choose not to attack in the case of high belief for his payoff, while in the case of low belief, he will choose different attack methods according to the cost of corresponding defense measures, which is in line with reality. The whole game process of the two sides presents a dynamic balance.

4.2. Influence of Detection Rate and False Alarm Rate on Prior Belief. Detection rate P_D and false alarm rate P_F have important effects on belief convergence. In MATLAB, we compared the changes of three groups of prior belief correction under different detection rates and false alarm rates,

Table 3: The main characteristics and selected features of different DDoS attacks.

DDoS attack	Characteristics	Features
SYN Flood	High speed, source address spoof, and destination address concentricity	FG, SG, and DG
ICMP Flood	Large amount of short time, high speed, and destination address concentricity	AP, AB, FG, and DG
UDP Flood	Heavy traffic in short time, high speed, source address spoof, and destination address concentricity	AP, AB, FG, SG, and DG

Table 4: DDoS classification results based on SVM.

DDoS attack	Training dataset (train/test)	Accuracy (%)
SYN Flood	5555/2778	99.93
ICMP Flood	7390/3697	100
UDP Flood	5882/2941	99.86

Table 5: Configurations of the desktop.

Equipment	Desktop
CPU	(R)Core (TM)i5-3470@3.20" title = "mailto:Intel (R)Core (TM)i5-3470@3.20">Intel (R)Core (TM)i5-3470@ 3.20 GHz
Memory size	16 GB
Operating system	Windows 10
Type of operating system	64 bits
Hard drive capacity	1 TB

Table 6: Configurations of the virtual machine.

Equipment	Virtual machine
CPU	(R)Core (TM)i5-3470@3.20" title = "mailto:Intel (R) Core (TM) i5-3470@3.20">Intel (R) Core (TM) i5-3470@ 3.20 GHz
Memory size	2 GB
Operating system	Ubuntu 16.04 LTS
Type of operating system	64 bits
Disk	40 GB

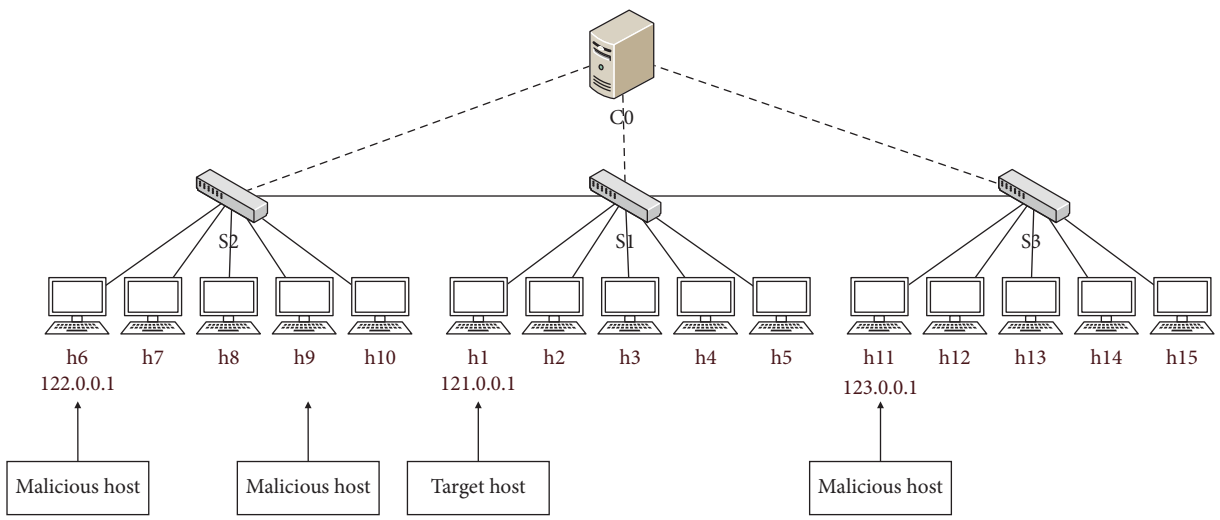

Figure 2: The test network topology.

and they are $P_D = 0.9$ and $P_F = 0.02$ (blue ○ line); $P_D = 0.7$ and $P_F = 0.1$ (black < line); $P_D = 0.5$ and $P_F = 0.2$ (red ∗ line). This is shown in Figure 4. As can be seen from Figure 4, in the case of high detection rate and low false alarm rate, prior belief converges more quickly, can be upregulated or downregulated quickly, and will not decline

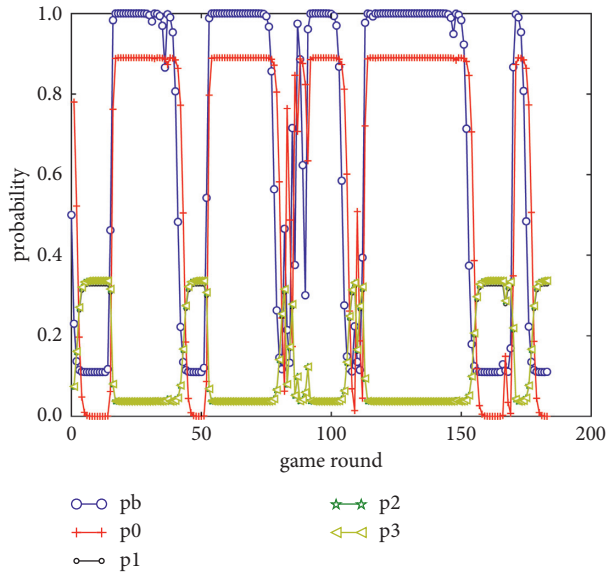

Figure 3: Modification of prior belief and attacker's behavior.

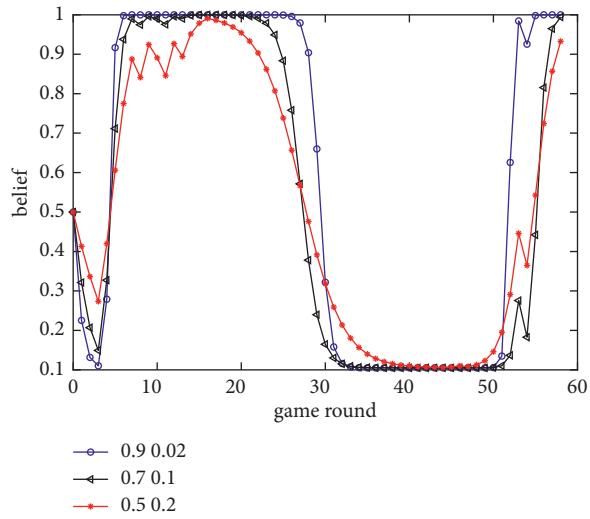

Figure 4: Prior belief correction under different detection rates and false alarm rates.

immediately after the attack but will remain for a longer period of time. In the case of low detection rate and high false alarm rate, belief convergence is slow and may not reach 1, which is not conducive to defense. Therefore, the selection of detection method has a very important impact on the performance of the whole system. The accuracy of SVM is very high, and it is competent for this work.

4.3. Traffic Data of the Targeted Host. During the game, real-time monitoring of traffic is carried out on the port of the targeted host to verify whether the attack and defense game is in a dynamic balance and whether the attacker's strategy is changing dynamically. The monitoring result is shown in Figure 5. Wireshark is used to detect the real-time traffic of the targeted host h1. The horizontal axis was the time, and the vertical axis was the packet speed. It can be clearly seen from Figure 5 that the traffic rate of the targeted host presents the characteristics of dynamic changes. When the defender has a high prior belief for the presence of the attacker, the probability of the attacker choosing not to attack increases greatly, and the traffic of the attacker is normal during this period. On the contrary, the packet rate of the targeted host increases significantly when attacked, almost tens of times more than normal.

4.4. Time Delay of the Targeted Host. Time delay is a very important reference index in the network, which represents the patency of the network. This paper collects the delay data of the ping operation of the normal host h2 against targeted host h1 in the game, as shown in Figure 6. It can be clearly seen from the figure that the ping delay increases significantly and exhibits periodic characteristics when the attack is happening.

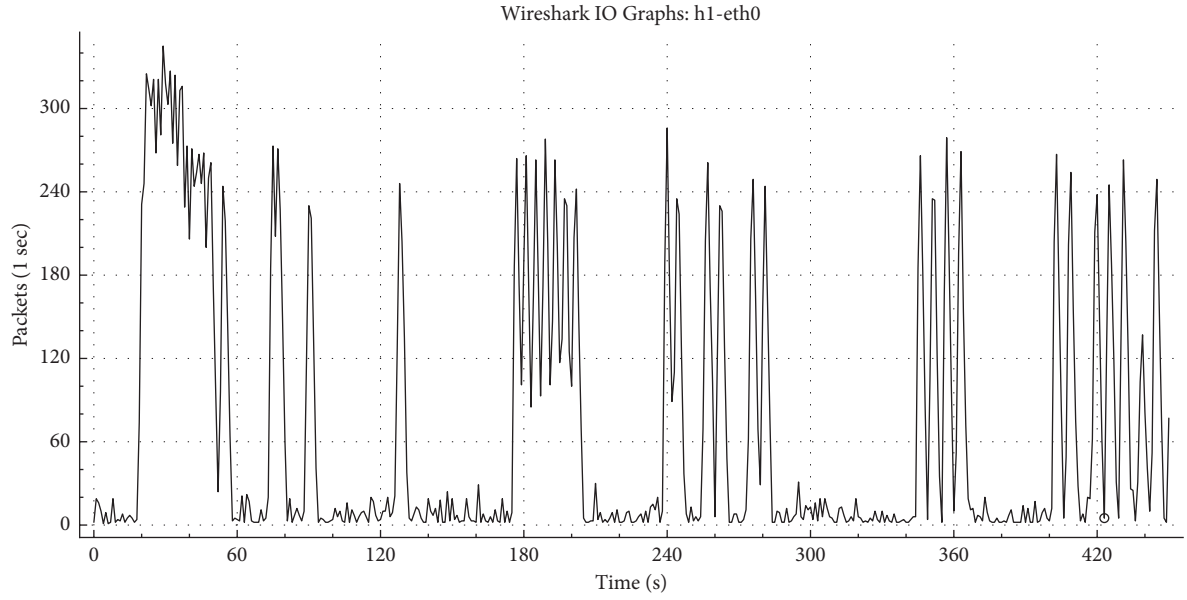

Figure 5: Traffic statistics of the targeted host.

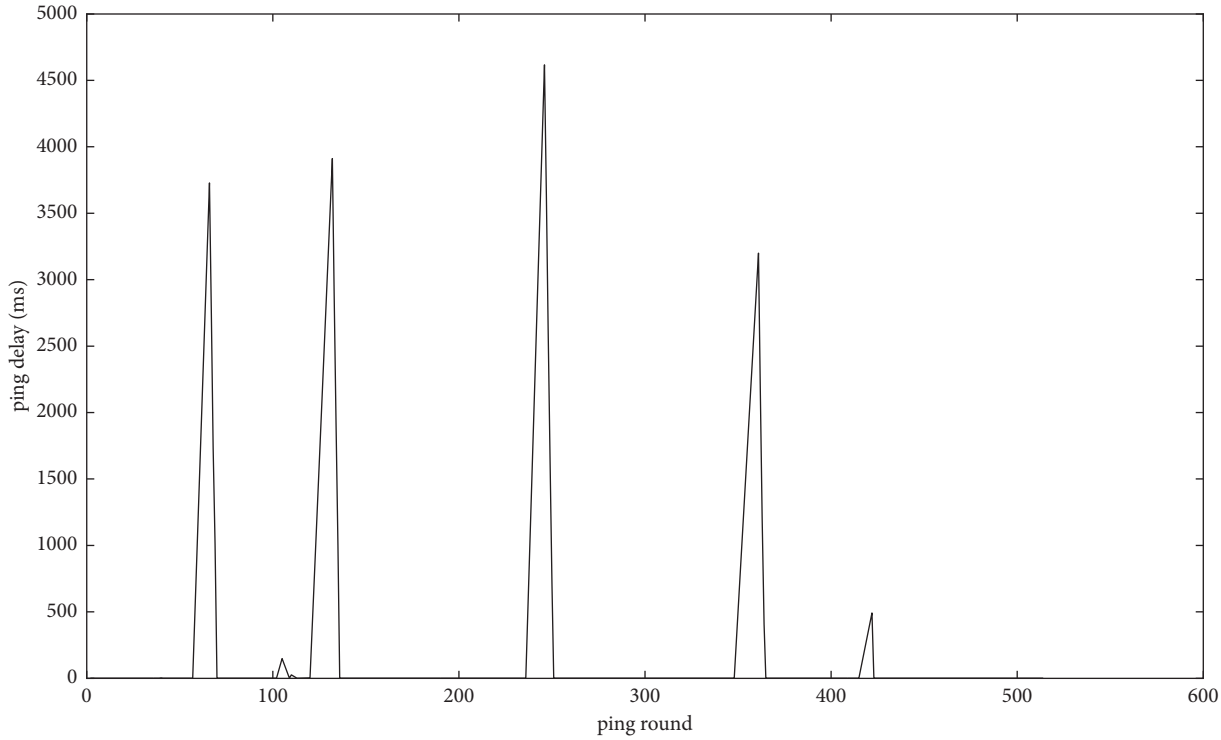

Figure 6: Time delay of the targeted host.

It is in line with expectations. However, due to the large traffic of DDoS attacks, some ICMP packets sent by h1 during the attack of h2 could not be echoed and were lost, resulting in the loss of data. Therefore, some delay peaks in Figure 6 are low, while the others are high. Under normal network conditions, the average echo time of ping is about 0.02~0.05 ms, but under DDoS attacks, it can be as high as several seconds, and the network is congested or even crashed.

4.5. Hardware Consumption. The main purpose of this paper is to reduce the defense cost and improve defense efficiency. To show the improvement, we choose an SVM method that covers all the five characteristics (AP, AB, FG, SG, and DG) as described in Section 3.5. The traditional SVM method is a method without game theory that does not differentiate each DDoS attack's cost. In this traditional SVM defense scheme, the cost of the defense system is maintained high all the time.

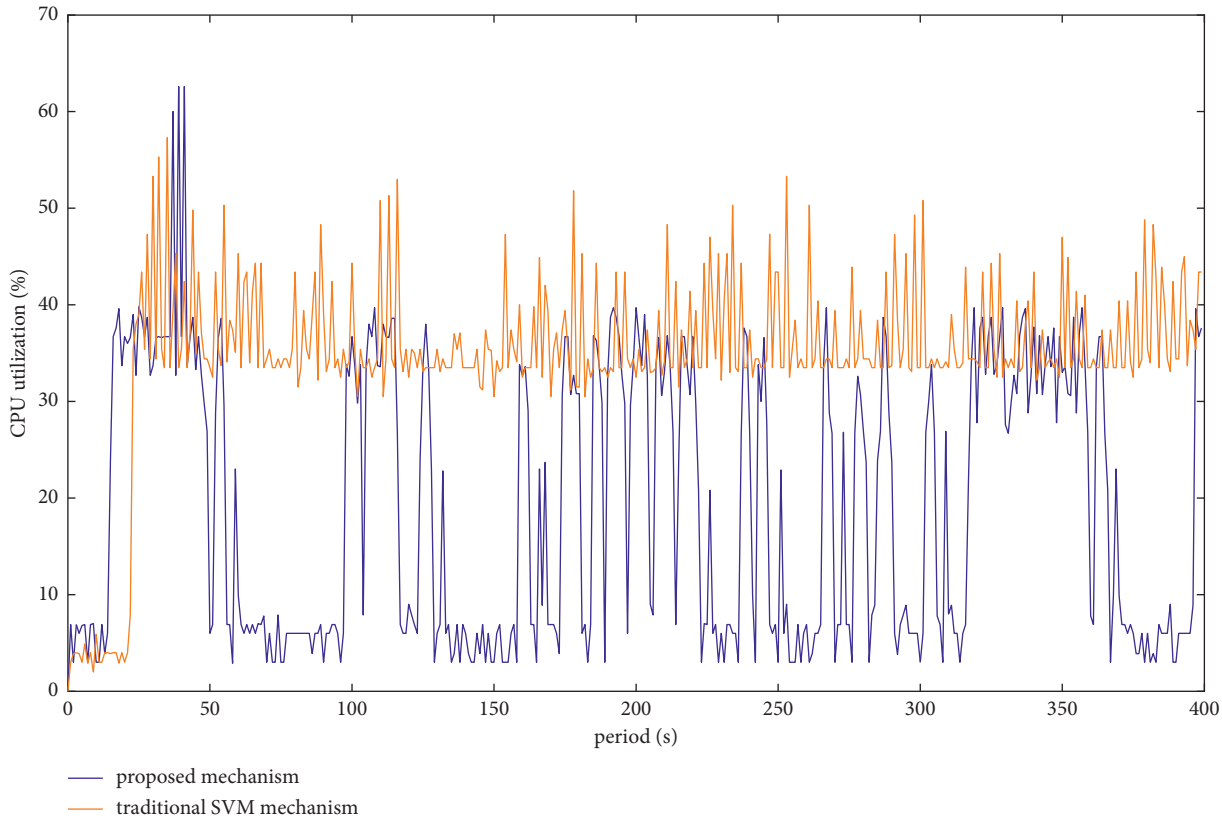

Figure 7: Comparison of the hardware consumption of this mechanic with that of a traditional mechanic.

We quantify the overhead in terms of the CPU usage of the controller process. As shown in Figure 7, the CPU utilization of our proposed mechanism against DDoS attacks is compared with that of the traditional SVM mechanism. The traditional mechanism is the machine learning method without game theory which does not differentiate each DDoS attack's cost. It can be seen that, in the case based on game theory, both sides of the attacker and defender are rational, and the attacker will measure its own payoffs and make the decision not to attack, so the defender does not have to maintain the defense all the time, and the whole process also presents a dynamic balance. However, under the traditional mechanism, both sides of the attack and defense will not rationally analyze the current situation and almost always maintain the most expensive attack and defense behavior, resulting in unnecessary waste. In this paper, we collected the changes in CPU utilization of the two mechanisms over a 400s period. The average CPU utilization of the traditional SVM mechanism (orange line) was 36.5%, while the average CPU utilization of the defense mechanism (blue line) was 20.6%, and the consumption was reduced by 43.6%, which reduced overhead.

5. Conclusion

In this paper, with the advantage of optimal strategy selection, a DDoS attack and defense model based on a dynamic Bayesian game is designed for the DDoS attack problem faced by SD-SAGIN. Different detection methods based on SVM are designed for different DDoS attack types. The detection accuracy is high, and both sides have four corresponding attack and defense methods. The defender adjusts his prior belief that there is an attacker in the network by detecting the network situation. Under Nash equilibrium, the defender randomly selects defense methods in fixed strategy space, and the attacker adjusts the selection probability of different attack strategies based on the change of prior beliefs. The two sides form a dynamic balance in the game process, which can effectively reduce the attack probability of the attacker and reduce the unnecessary resource waste of the defender. Compared with the traditional SVM defense method, the proposed method can reduce defense overhead and improve defense efficiency while ensuring network security. However, while reducing the overhead, the detection delay of DDoS is increased to a low degree, which needs to be improved in the subsequent work.

Data Availability

All data, models, and code generated or used during the study are included in the paper.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the State Key Laboratory of Integrated Services Networks, Xidian University (ISN22-13), and the Higher Education Department of the Ministry of Education Industry-university Cooperative Education Project (201802007013).

References

[1] N. Kato, Z. M. Fadlullah, F. Tang et al., "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 140–147, 2019.

[2] R. Ferrús, H. Koumaras, O. Sallent et al., "SDN/NFV-enabled satellite communications networks: opportunities, scenarios and challenges," *Physical Communication*, vol. 18, no. P2, pp. 95–112, 2015.

[3] N. Dayal, P. Maity, S. Srivastava, and R. Khondoker, "Research trends in security and DDoS in SDN," *Security & Communication Networks*, vol. 9, 2016.

[4] R. Swami, M. Dave, and V. Ranga, "Defending DDoS against software defined networks using entropy," in *Proceedings of the 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pp. 1–5, Ghaziabad, India, April 2019.

[5] A. Ahalawat, S. S. Dash, A. Panda, and K. S. Babu, "Entropy based DDoS detection and mitigation in OpenFlow enabled SDN," in *Proceedings of the 2019 International Conference on Vision towards Emerging Trends in Communication and Networking (ViTECoN)*, pp. 1–5, Vellore, India, March 2019.

[6] Y. Xu, H. Sun, F. Xiang, and Z. Sun, "Efficient DDoS detection based on K-FKNN in software defined networks," *IEEE Access*, vol. 7, pp. 160536–160545, 2019.

[7] Y. Chen, J. Pei, and D. Li, "DETPro: a high-efficiency and low-latency system against DDoS attacks in SDN based on decision tree," in *Proceedings of the ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, May 2019.

[8] P. Preamthaisong, A. Auyporntrakool, P. Aimtongkham, T. Sriwuttisap, and C. So-In, "Enhanced DDoS detection using hybrid genetic algorithm and decision tree for SDN," in *Proceedings of the 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 152–157, Chonburi, Thailand, July 2019.

[9] S. Haider, A. Akhunzada, G. Ahmed, and M. Raza, "Deep learning based ensemble convolutional neural network solution for distributed denial of service detection in SDNs," in *Proceedings of the 2019 UK/China Emerging Technologies (UCET)*, pp. 1–4, Glasgow, UK, August 2019.

[10] Z. Liu, Y. He, W. Wang, and B. Zhang, "DDoS attack detection scheme based on entropy and PSO-BP neural network in SDN," *China Communications*, vol. 16, no. 7, pp. 144–155, 2019.

[11] B. Celesova, J. Val'ko, R. Grezo, and P. Helebrandt, "Enhancing security of SDN focusing on control plane and data plane," in *Proceedings of the 2019 7th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 1–6, Barcelos, Portugal, June 2019.

[12] X. Tang, Q. Zheng, J. Cheng, V. Sheng, R. Cao, and M. Chan, "A DDoS attack situation assessment method via optimized cloud model based on influence function," *Computers, Materials & Continua*, vol. 58, no. 2, pp. 1263–1281, 2019.

[13] M. V. O. De Assis, A. H. Hamamoto, T. Proenca, and M. L. Proença, "A game theoretical based system using holt-winters and genetic algorithm with fuzzy logic for DoS/DDoS mitigation on SDN networks," *IEEE Access*, vol. 5, pp. 9485–9496, 2017.

[14] B. Wang, Y. Sun, and X. Xu, "A scalable and energy-efficient anomaly detection scheme in wireless SDN-based mMTC networks for IoT," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1388–1405, 2021.

[15] M. Du and K. Wang, "An SDN-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial Internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 648–657, 2020.

[16] A. Chowdhary, S. Pisharody, A. Alshamrani, and D. Huang, "Dynamic game based security framework in SDN-enabled cloud networking environments," in *Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization—SDN-NFVSec'17*, pp. 53–58, Dallas, TX, USA, March 2017.

[17] R. A. Niazi and Y. Faheem, "A bayesian game-theoretic intrusion detection system for hypervisor-based software defined networks in smart grids," *IEEE Access*, vol. 7, pp. 88656–88672, 2019.

[18] Z. Tu, H. Zhou, K. Li, M. Li, and A. Tian, "An energy-efficient topology design and DDoS attacks mitigation for green software-defined satellite network," *IEEE Access*, vol. 8, pp. 211434–211450, 2020.

[19] K. Li, H. Zhou, Z. Tu, W. Wang, and H. Zhang, "Distributed network intrusion detection system in satellite-terrestrial integrated networks using federated learning," *IEEE Access*, vol. 8, pp. 214852–214865, 2020.

[20] K. Guo, D. Wang, H. Zhi, Y. Lu, and Z. Jiao, "A trusted resource-based routing algorithm with entropy estimation in integrated space-terrestrial network," *IEEE Access*, vol. 8, pp. 122456–122468, 2020.

[21] C. Guo, C. Gong, J. Guo, Z. Wei, Y. Han, and S. Zaman Khan, "Software-defined space-air-ground integrated network architecture with the multi-layer satellite backbone network," *Computers, Materials & Continua*, vol. 64, no. 1, pp. 527–540, 2020.