Methods for blocking malicious traffic with static Bayesian game in IIoT

Maoli Wang(✉), Bowen Zhang and Quanxi Xia

School of Cyber Science and Engineering, Qufu Normal University, Qufu 273165, China
wangml@qfnu.edu.cn

Abstract. With the development of the Industrial Internet of Things (IIoT), its security issues have become prominent, and network attacks have continued to increase, including malicious traffic threats. There are already many effective methods for detecting malicious traffic in the Industrial Internet of Things, but how to handle detected malicious traffic lightly and effectively? Based on this problem, we propose a method to dynamically adjust malicious traffic in the Industrial Internet of Things using Software-Defined Networking (SDN). With the help of SDN's programmability of the network and the characteristics of decoupling the control plane and the data plane, through the SDN controller OpenFlow rule entries corresponding to malicious traffic are generated, and then the SDN switch updates the flow table to achieve the purpose of blocking malicious traffic. At the same time, we consider two types of known types of malicious traffic and unknown types of malicious traffic. Different strategies of traffic blocking, including traffic dropping and traffic redirection, conduct a static Bayesian game between two types of malicious traffic and two traffic blocking strategies, taking into account factors such as current and future benefits, response costs, and risk levels, through the Harsanyi transformation reasoning proves that the Nash equilibrium point and equilibrium strategy are found, and then the strategy is numerically analyzed and experimentally verified. The final result is that when the known type of malicious traffic is discarded and the unknown type of malicious traffic is redirected, comprehensive maximum utility.

Keywords: IIoT, blocking malicious traffic, static Bayesian game, SDN.

1 Introduction

After the global announcement about the fourth industrial revolution, China proposed the 2025 industrial manufacturing strategy [1]. Although the IIoT has flourished in recent years, it has also been subject to an increasing number of cyberattacks and malicious behaviors, leading to the leakage of sensitive information, damage to industrial infrastructure, and economic losses [2]. For the ever-evolving and dynamic Industrial IoT systems, the use of popular static defense measures (such as passwords, encryption or firewalls) may not be effective [3]. Therefore, active defense is more suitable for Industrial IoT security defense [4]. Existing active defense Strategies include honeypots, redirection, containment and other methods [5], which can indirectly improve the security of the system by keeping attackers away from the real system [6]. This article mainly focuses on proactive defense against malicious traffic in security issues. Since

the previous research on malicious traffic detection has been relatively complete, we adopt the method of blocking known types and unknown types of malicious traffic to improve industrial physical security. The security factor of networked systems.

The characteristics of network offensive and defensive games in the Industrial Internet of Things are mainly manifested in six aspects: goal opposition, strategic dependence, non-cooperative relationships, incomplete information, dynamic evolution, and interest-driven [7]. Game theory mainly studies the strategic choices of participants with interdependent behaviors, which strategy to choose depends on the benefit value of the strategy. [8]The utility of the strategy is one of the important basis for each participant in game theory to make rational decisions. [9]Game theory can screen out the strategies for the participants through theoretical analysis and research. [10,11]The decision with the highest return. Considering that in the industrial Internet of Things environment, the information mastered by both the attacker and defender is incomplete and the decision-making behaviors of the attackers and defenders occur logically at the same time, we model the malicious traffic defense process as a static Bayesian game model. The two sides of the game are malicious traffic and traffic blocking strategies respectively. Malicious traffic includes known types and unknown types. Traffic blocking strategies include traffic dropping and traffic redirection. The utility considers three factors, namely response cost and risk level and impact on current and future earnings.

The premise of traffic dropping and traffic redirection is the programmability of the network, which is exactly in line with the characteristics of SDN. [12]SDN provides the ability to program the network through centralized network control and decouples the control plane and data plane [13], based on Programming can enhance network security through rapid traffic drainage [14]. The key to SDN's programmability lies in the OpenFlow network communication protocol. OpenFlow is a southbound application programming interface (API). The OpenFlow switch consists of one or more flow tables composed of various rules called flows, and performs packet lookups. Consisting of forwarding group tables and OpenFlow channels to external controllers, each flow matches a specific set of packets and performs operations on them [8]. Therefore, we use the SDN controller to generate corresponding OpenFlow rule entries from the malicious traffic information, and then publish them to the SDN switch to update the flow table to achieve the purpose of traffic discarding and traffic redirection.

To solve the threat of malicious traffic in the Industrial Internet of Things, rather than just detecting malicious traffic, we proposed a method for blocking malicious traffic with static Bayesian game in IIoT, which plays a static Bayesian game between malicious traffic and traffic blocking strategies, malicious traffic includes known types and unknown types, traffic blocking strategies include traffic dropping, traffic redirection. The utility considers the three factors of response cost, risk level, and impact on current and future revenue, and finds a balanced strategy through inference proof, that is, traffic is dropped for known types of malicious traffic, and traffic is redirected for unknown types of malicious traffic. For the implementation of these two traffic blocking strategies, we use the programmability of SDN to dynamically adjust the malicious traffic. The SDN controller generates OpenFlow rule entries corresponding to malicious traffic, and then the SDN switch updates the flow table to achieve the purpose of blocking

malicious traffic. In summary, we proposed a SDN method to dynamically adjust industrial IoT malicious traffic based on static Bayesian game. The main contributions of this paper are as follows:

1) We propose methods for blocking malicious traffic with static Bayesian game in IIoT, which finds the best defense strategy by letting malicious traffic attack and defense strategies perform a static Bayesian game.

2) We adopt different traffic blocking strategies for known types and unknown types of malicious traffic respectively, and consider the three influencing factors of current and future revenue, response cost, and risk level in selecting defense strategies.

3) We use the programmability of SDN and the characteristics of decoupling the control plane and the data plane to block malicious traffic in the IIoT.

The remainder of this article is organized as follows:

Section 2 summarizes IIoT active defense methods, the use of game theory in IIoT attack and defense issues, and related work on using SDN to solve IIoT security issues. Section 3 describes the proposed methods for blocking malicious traffic with static Bayesian game in IIoT and SDN-based traffic blocking strategy. Section 4 is the experimental environment, results, analysis, and comparative experiments. Finally, Section 5 gives conclusions.

2 Related Work

In IIoT security issues, traditional static security-based environments such as passwords or firewalls may be low-cost solutions to mitigate simple attacks, but they do not provide sufficient evidence against more complex attacks [3], but, Active defense can effectively deal with security issues. There have been few previous studies on active defense against IIoT malicious traffic, but there are a lot of studies using honeypots for active defense. Honeypot technology is an important means of active defense, by setting simulation targets, decoy malicious attacks, realize the capture of malicious behaviors and data, and provide effective data for security analysis. Literature [3] proposes a cloud-based active defense method for IoT network attacks, "CICADA", which uses three simulated deception environments (Honeynet, Pseudocomb and Honeyclone) to deceive attackers. Literature [15] proposed a dynamic bounded rational honeypot APT game model (DBHM) method, which collects advanced persistent threats (APTs) attack information by deploying honeypots in IIoT and then regains control of the attacked IIoT server to reduce IIoT security risks. Literature [16] combines honeypots with machine learning to detect botnets, and ensures the security of IoT devices by closely monitoring and acquiring the attack behaviors of botnets.

At the same time, due to the rise of game theory in network attack and defense, it has also been used to deal with IIoT security issues, whether it is complete information games, incomplete information games, or new types of games, such as differential games, evolutionary games, random games, etc., are all used into industrial IoT security. Holmgren et al. [17] modeled the offensive and defensive confrontation process in the power grid as an offensive and defensive game model based on game theory, and studied the performance of different defense strategies under different target situations.

Ziad et al. [18] studied the security protection strategy of the smart grid. They constructed a smart grid attack and defense game model based on static game theory and discussed how to select the most cost-effective security protection strategy. Yang et al. [19] aimed at the optimal defense strategy selection problem in the Internet of Things environment, combined with the characteristics of the Internet of Things, proposed a multi-stage network attack and defense game model, and designed a defense strategy selection algorithm. Chen et al. [20] proposed a multi-stage attack and defense signal game model to solve the problem of optimal strategy selection for industrial control systems to defend against phishing attacks. They used symbolic variables to quantify attack and defense benefits, gave the optimal strategy selection method, and analyzed Key factors affecting the outcome of the game. Literature [21] uses the Bayesian game method to defend against link flooding attack (LFA) in the Internet of Things, using a single-round defensive decision game to find the defender's optimal strategy.

SDN's programmability of the network and its characteristics of decoupling the control plane and data plane enable SDN to effectively solve network security problems, and is also applicable to the IIoT [22]. Literature [23] proposed a new secure IoT framework based on SDN, which can use session IP counters and IP payload analysis to detect vulnerabilities in IoT devices or malicious traffic generated by IoT devices. Literature [24] proposed an AMLSDM framework. With the support of an adaptive machine learning classification model, an SDN-supported security mechanism was developed for IoT devices, using remote SDN controllers to mitigate detected open flows (OF) DDoS attacks on the switch and reconfigure network resources for legitimate network hosts. Literature [25] proposed a method of using SDN to protect IoT devices and HTTP to mitigate and prevent security attacks without modifying IoT devices.

3 Methodology

Generally speaking, our method is the SDN dynamic adjustment method for IIoT malicious traffic based on the static Bayesian game. The overall structure is divided into two parts. The first part is the static Bayesian game model. We build the IIoT malicious traffic defense process. The model is a static Bayesian game model. The two sides of the game are industrial Internet of Things malicious traffic and traffic blocking strategies. After establishing the game model, Harsanian transformation is performed, and finally, the Bayesian Nash equilibrium point is proved by reasoning. The second part is the traffic blocking strategy based on SDN. We use the SDN controller to generate OpenFlow rule entries corresponding to malicious traffic, and then the SDN switch updates the flow table, ultimately achieving the purpose of blocking malicious traffic. Figure 1 shows the overall framework of our method.

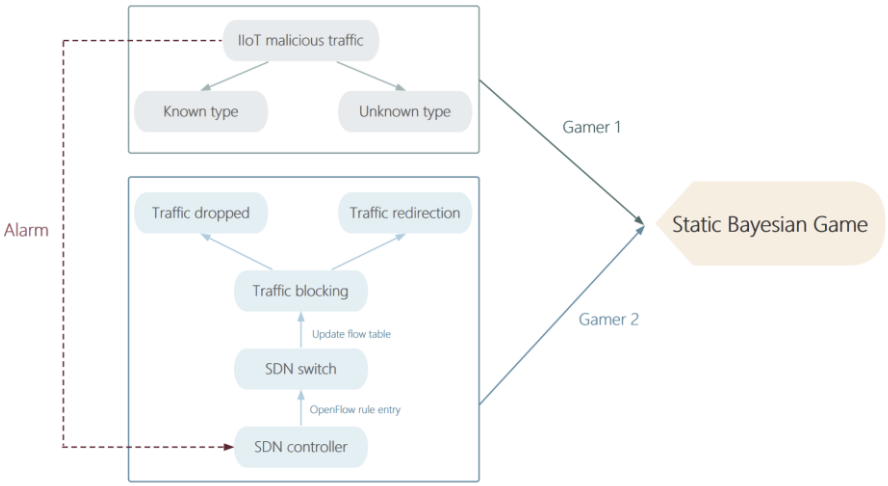

Fig. 1. Framework of IIoT malicious traffic blocking methods

3.1 Static Bayesian Game Model

Model building. The key factor in static Bayesian games is that each player knows its own utility function, but cannot exactly know the utility functions of other players. [26]During the game, in order to transform the lack of understanding of the benefits into the lack of understanding of the types, we perform the following conversion [27].

[28]If t_i is used to represent the type of game party i, T_i is used to represent the type space of game party i, $t_i \in T_i$, and $u_i = u_i(a_1, \dots, a_n, t_i)$ is used to represent the utility of game party i under the strategy combination $(a_1, \dots, a_n)$, each type t_i corresponds to possible situations of different utility functions for player i, whose values are known to player i but not to other players, reflecting the characteristics of incomplete information in static Bayesian games. [29]Therefore, the general expression of the static Bayesian game is[30]:

$$G = \{A_1, \dots, A_n; T_1, \dots, T_n; u_1, \dots, u_n\} \tag{1}$$

Among them, A_i is the strategy space of player i, T_i is the type space of player i, $u_i = u_i(a_1, \dots, a_n, t_i)$ is the utility of player i, which is the strategy combination $(a_1, \dots, a_n)$ and functions of type t_i.

The type of player i, as the private information of player i, determines the utility function $u_i(a_1, \dots, a_n, t_i)$ of player i. The inference of player i is expressed as $p_1, \dots, p_n$, and the inference of player i is $p_i = p_i\{t_{-i}|t_i\}$ describes its uncertainty about the possible types t_{-i} of the other n-1 players given its own type t_i. Therefore, the expression of the static Bayes game is updated to:

$$G = \{A_1, \dots, A_n; T_1, \dots, T_n; p_1, \dots, p_n u_1, \dots, u_n\} \tag{2}$$

Harsanyi transformation. To further transform the static game with incomplete information into a dynamic game with complete but imperfect information, we introduce the Harsanyi transformation. The specific steps are:

1) Introduce a virtual game party "Nature", which can be called gamer 0. It randomly extracts its own type for each actual game party, that is, randomly assigns types to the game parties. These types constitute the type vector $t = (t_1, \dots, t_n)$.

2) "Nature" only lets each player know his own type, but not other players.

3) All gamers select actions at the same time, that is, each actual gamer selects action plans $(a_1, \dots, a_n)$ from their respective behavior spaces at the same time.

4) Except for gamer 0, which is "Nature", the other gamers each obtain utilities $u_i(a_1, \dots, a_n, t_i)$.

The above-mentioned converted game is a dynamic game, because this game has an obvious time sequence, that is, there are two stages of choices. [31]First, the choice of the "Nature"; then, the simultaneous choices of gamers $1, \dots, n$. At least some of the players do not fully understand the consequences of the type that "Nature" chooses for the other players in the first stage. Therefore, this is a dynamic game with imperfect information. When the "Nature" selection direction is used to represent the type of actual gamers, then under each game strategy combination $(a_1, \dots, a_n, t_i)$, the gains of each gamer are $u_i(a_1, \dots, a_n, t_i)$ is determined and known to each gamer. [32]Obviously, this is a complete information game. At this time, the original incomplete information game becomes a complete information game.

The above symbols and their meanings are shown in Table 1.

Table 1. Symbols and meanings

Symbol	Meaning
t_i	Type of gamer i
T_i	Type space of gamer i
a_i	Strategy combination of gamer i
A_i	Strategy space of gamer i
u_i	Utility of gamer i under a_i

Bayesian Nash Equilibrium. Since the static Bayesian game can be regarded as a dynamic game in which the type of each player is first "Nature" selected, and then each gamer selects a strategy at the same time, a strategy for each gamer in the static Bayesian game is, It is their complete plan on how to choose among their various possible types, that is, for static Bayesian games $G = \{A_1, \dots, A_n; T_1, \dots, T_n; p_1, \dots, p_n; u_1, \dots, u_n\}$, a strategy of gamer i is a function $S_i(t_i)$ of its various possible types $t_i (t_i \in T_i)$. That is to say, for various types t_i, "Nature" extracted for gamer i in T_i, $S_i(t_i)$contains the corresponding action a_i selected by gamer i from its own behavior space A_i.

It can be seen that the strategy of the gamer in the static Bayesian game is a function of the type space and the behavior space. All such functions constitute the strategy space of the gamer, that is, the feasible strategy set $S_i(t_i)$ of the gamer i is the definition

domain. is T_i, the set of all possible functions whose value range is A_i. Since there are many functional relationships between sets, if no restrictions are imposed, the strategy space of the gamers in static Bayesian games is often very large, with many or even infinite elements. Depending on the different situations of the strategy function $S_i(t_i)$, the actions a_i determined by them for different types can be different or the same.

Therefore, in the static Bayesian game $G = \{A_1, \dots, A_n; T_1, \dots, T_n; p_1, \dots, p_n; u_1, \dots, u_n\}$, if for any game party i and each of its possible types $t_i \in T_i$, the action a_i selected by $S_i^*(t_i)$ can all satisfy:

$$\max_{a_i \in A_i} \sum_{t_{-i}} \{u_i[S_1^*(t_1), \dots, S_{i-1}^*, a_i, S_{i+1}^*(t_{i+1}), \dots, S_n^*(t_n), t_i] p(t_{-i}|t_i)\} \tag{3}$$

Then the strategy combination $S^* = (S_1^*, \dots, S_n^*)$ of the game is called a pure strategy Bayesian Nash equilibrium of G. [33]This definition shows that when a strategy combination of players in a static Bayesian game is a Bayesian Nash equilibrium, no player wants to change their strategy, which is completely consistent with the connotation of Nash equilibrium [34].

Scene substitution. For our usage scenario, the two gamers in the game are attacker 1, which is T_1, and defender 2, which is T_2. We choose the IIoT malicious traffic attack t_1 in T_1 and the IIoT traffic blocking strategy t_2 in T_2 to play the game. The policy space A_1 of t_1 includes the known type of malicious traffic attack $a_{(1,1)}$ and the unknown type of malicious traffic attack $a_{(1,2)}$, and the policy space A_2 of t_2 includes the traffic drop policy $a_{(2,1)}$ and traffic redirection strategy $a_{(2,2)}$, $u_{(i,j)}$ is t_i's utility under $a_{(i,j)}$. Among them, $u_{(1,j)}$ includes attack cost $c_{(1,j)}$, damage effect $e_{(1,j)}$ and attack persistence d_j, $u_{(2,j)}$ includes response cost $c_{(2,j)}$, current and future returns $e_{(2,j)}$, risk level r_j. The definition of $u_{(i,j)}$ is as follows:

$$u_{(1,j)} = e_{(1,j)} + d_j - c_{(1,j)} \tag{4}$$

$$u_{(2,j)} = e_{(2,j)} - c_{(2,j)} - r_j \tag{5}$$

The above symbols and their meanings are shown in Table 2.

Table 2. Symbols and meanings

Symbol	Meaning
T_1	Attacker
t_1	IIoT malicious traffic attack
T_2	Defender
t_2	IIoT traffic blocking strategy
A_1	The strategy space of t_1
$a_{(1,1)}$	The first strategy in A_1, known types of malicious traffic attacks
$a_{(1,2)}$	The second strategy in A_1, unknown type of malicious traffic attack
A_2	The strategy space of t_2
$a_{(2,1)}$	The first strategy in A_2, the traffic dropped strategy
$a_{(2,2)}$	The second strategy in A_2, the traffic redirection strategy
$u_{(i,j)}$	The utility of the i-th gamer in the j-th strategy
$e_{(i,j)}$	The current and future benefits of the i-th gamer in the j-th strategy
$c_{(i,j)}$	The response cost of the i-th gamer in the j-th strategy
r_j	The risk level of strategy j

After introducing the virtual gamer "Nature" through Harsanyi transformation, the Bayesian game tree formed is shown in Figure 2, and the corresponding utility matrix is shown in Figure 3, where U_i $(i = 1,2 \ldots 6)$ is represented by $u_{(1,j)}$ and $u_{(2,j)}$, reresents the overall return under the i-th strategy.

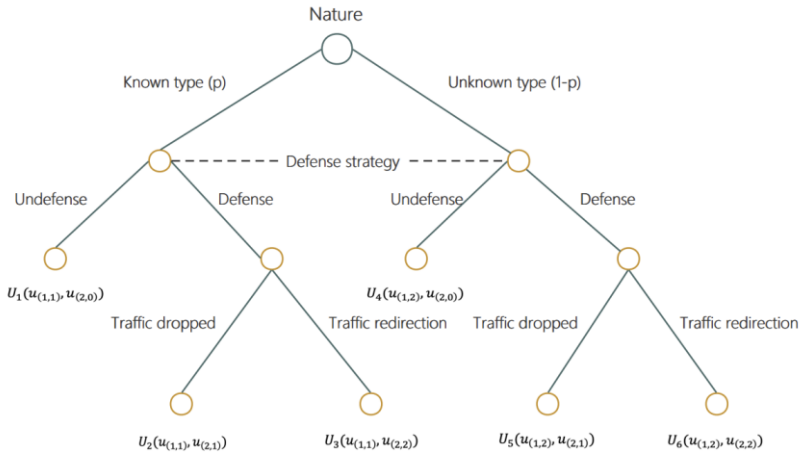

Fig. 2. Bayesian game tree

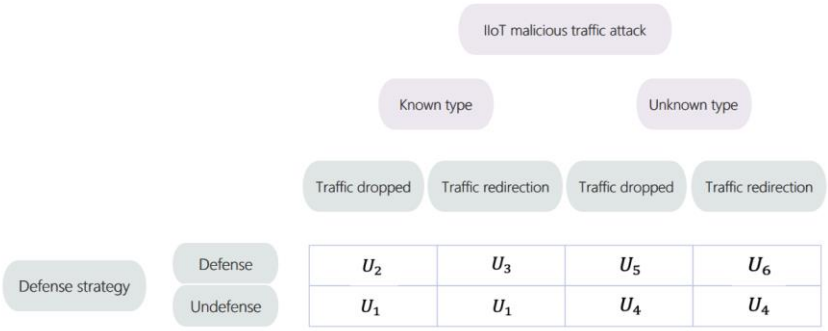

Fig. 3. Utility matrix

Since we mainly consider the overall utilities of gamer 2 when performing traffic discarding and traffic redirection for known types and unknown types of IIoT malicious traffic attacks respectively, we evaluate the overall utilities of gamer 2 in four scenarios (U_2, U_5), (U_2, U_6), (U_3, U_5), (U_3, U_6). The income b_i is analyzed:

$$\begin{cases} b_1 = U_2(u_{(2,1)}) \cdot p + U_5(u_{(2,1)}) \cdot (1-p) \\ b_2 = U_2(u_{(2,1)}) \cdot p + U_6(u_{(2,2)}) \cdot (1-p) \\ b_3 = U_3(u_{(2,2)}) \cdot p + U_5(u_{(2,1)}) \cdot (1-p) \\ b_4 = U_3(u_{(2,2)}) \cdot p + U_6(u_{(2,2)}) \cdot (1-p) \end{cases} \tag{6}$$

Substitute into formula (5):

$$\begin{cases} b_1 = U_2(e_{(2,1)} - c_{(2,1)} - r_1) \cdot p + U_5(e_{(2,1)} - c_{(2,1)} - r_1) \cdot (1-p) \\ b_2 = U_2(e_{(2,1)} - c_{(2,1)} - r_1) \cdot p + U_6(e_{(2,2)} - c_{(2,2)} - r_2) \cdot (1-p) \\ b_3 = U_3(e_{(2,2)} - c_{(2,2)} - r_2) \cdot p + U_5(e_{(2,1)} - c_{(2,1)} - r_1) \cdot (1-p) \\ b_4 = U_3(e_{(2,2)} - c_{(2,2)} - r_2) \cdot p + U_6(e_{(2,2)} - c_{(2,2)} - r_2) \cdot (1-p) \end{cases} \tag{7}$$

Expand further:

$$\begin{cases} b_1 = [U_2(e_{(2,1)}) - U_2(c_{(2,1)}) - U_2(r_1)] \cdot p + [U_5(e_{(2,1)}) - U_5(c_{(2,1)}) - U_5(r_1)] \cdot (1-p) \\ b_2 = [U_2(e_{(2,1)}) - U_2(c_{(2,1)}) - U_2(r_1)] \cdot p + [U_6(e_{(2,2)}) - U_6(c_{(2,2)}) - U_6(r_2)] \cdot (1-p) \\ b_3 = [U_3(e_{(2,2)}) - U_3(c_{(2,2)}) - U_3(r_2)] \cdot p + [U_5(e_{(2,1)}) - U_5(c_{(2,1)}) - U_5(r_1)] \cdot (1-p) \\ b_4 = [U_3(e_{(2,2)}) - U_3(c_{(2,2)}) - U_3(r_2)] \cdot p + [U_6(e_{(2,2)}) - U_6(c_{(2,2)}) - U_6(r_2)] \cdot (1-p) \end{cases} \tag{8}$$

3.2 SDN-based traffic blocking strategy

OpenFlow switch. The function of the OpenFlow switch is to parse and match the received data packets and then process them accordingly. The structure diagram is shown in Figure 4. The OpenFlow switch mainly includes flow tables, group tables, meters, and secure channels [35].

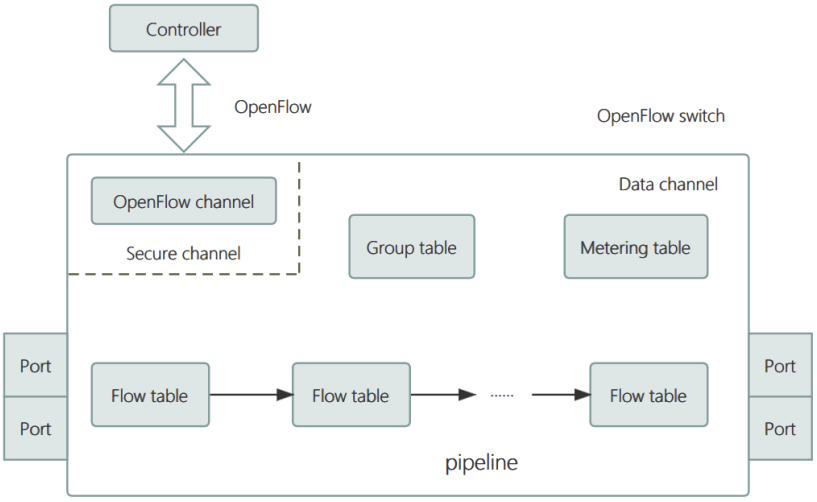

Fig. 4. OpenFlow structure diagram

The flow table generally contains several flow table items, which are used to match, process and forward data packets. The structure of the flow table items consists of matching fields, priorities, counters, instruction sets, timeouts and cookies [36].

Group table types are divided into required types and optional types. Required types are: ALL, Indirect. ALL means that no selection is required to perform the operation, and only all operations in the action bucket need to be executed. Indirect means that only one action in the action bucket is executed. Optional types are: Select, Fast failover [37].

The structure of metering table items includes metering ID, metering bandwidth and counter. [38]Among them, metered bandwidth specifies the bandwidth rate and processing behavior of data packets. The meter is directly connected to the flow meter item and is used as an optional instruction in the instruction set of the flow meter to provide quality-of-service operations for each flow measurement [39].

As can be seen from Figure 4, the secure channel is the channel for bidirectional communication between the OpenFlow switch and the external controller. The matching process of data packets by the OpenFlow switch is shown in Figure 5. It will be explained in detail below: the flow table must contain a set of flow table items to match the message fields or instructions entering or leaving the device. After the OpenFlow switch receives the data packet. For parsing and matching, the comparison will start from the first flow entry, and the flow entries will be checked according to the priority. If there is no matching option for the data packet in the first flow entry, the table pipeline will continue in sequence. Matching check, matching flow table N (flow table N is the last flow entry), if a matching entry is found, an action (forwarding, discarding, filtering, etc.) is performed. If the matching is not successful, the operation executes the configuration on the table-miss stream entry.

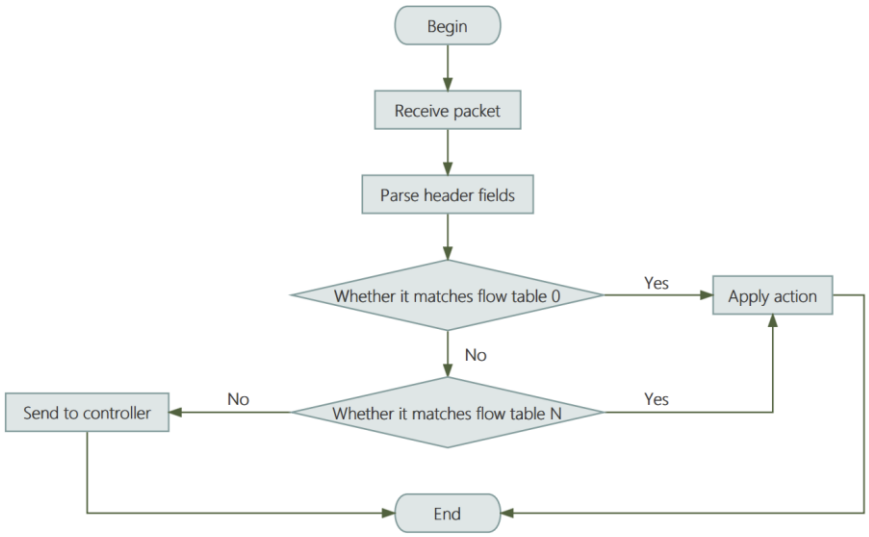

Fig. 5. The matching process of data packets by OpenFlow switches

OpenFlow Switch Specification. The development of OpenFlow provides convenience for the controller to monitor network traffic. [40]The controller manages the data forwarding of the switch, and the switch also submits status information to the controller through this protocol. OpenFlow has strict definition standards. OpenFlow messages can be mainly divided into three categories, namely Controller-to-Switch messages, Asynchronous messages and Symmetric messages [41].

The Controller-to-Switch message is a message type initiated by the controller to communicate with the switch. In this message type, the status of the switch can be viewed and the switch can be monitored. Asynchronous messages are the type of messages the switch sends to the controller. Symmetric messages are the type of messages that the controller and switch send to each other.

The interaction process between the controller and the OpenFlow switch:

1) The switch establishes a TCP connection with the controller.

2) The Hello message is sent from Open vSwitch (OVS) to the controller, and the data path between the controller and the OpenFlow switch is initialized.

3) The next message is Feature-request. The controller obtains the basic information characteristics of the switch to find the OVS Data Path ID (DPID) and locate the SDN Switch. The switch sends a Feature-reply message to indicate that the information has been received, and its response is called Feature-reply. The DPID is the MAC address of the switch plus a 16-bit address determined by the vendor. The controller will then send a message to obtain information from the switch.

4) Packet-in and Packet-out messages are used when the controller does not know the host MAC address. The ARP request is transmitted in a Packet-in message. In the IPv4 scenario, the ARP Reply is encapsulated in the Packet-out message. The IPv6 scenario works similarly to using the Neighbor Discovery Protocol (NDP).

5) Send Echo request and Echo response OpenFlow messages to maintain and verify the activity of the Controller-OVS connection.

4 Experiments

4.1 Experimental environment

Our experiments used the Mininet simulation platform. Mininet is a free and open-source network simulation framework based on Python [42]. It is also a simulation tool for building and testing software-defined networks (SDN) and network function virtualization (NFV). We use Iperf to generate three types of traffic in a random pattern, namely normal traffic, known types of malicious traffic, and unknown types of malicious traffic. The generated traffic conforms to the IIoT traffic characteristics. Other network configurations in the experiment are shown in Table 3.

Table 3. Experimental configuration

Equipment	Virtual machine
CPU	Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz
Operating System	CentOS-7.6
Network simulation platform	Mininet
SDN switch	Open vSwitch 2.5.0
SDN controller	Ryu 4.34
Southbound interface protocol	OpenFlow 1.3.0

4.2 Experimental results and analysis

We use the benefit b_j in the four scenarios in formula (8) as the evaluation index of our method.

Through experiments, we summarized the quantitative relationship between the three variables $e_{(2,j)}$、$c_{(2,j)}$、r_j in formula (8) according to the scene change. Taking $U_2(e_{(2,1)})$, $U_2(c_{(2,1)})$ and $U_2(r_1)$ in scenario 1 as the benchmark, represent the variables in the remaining scenarios, as shown in Figure 6.

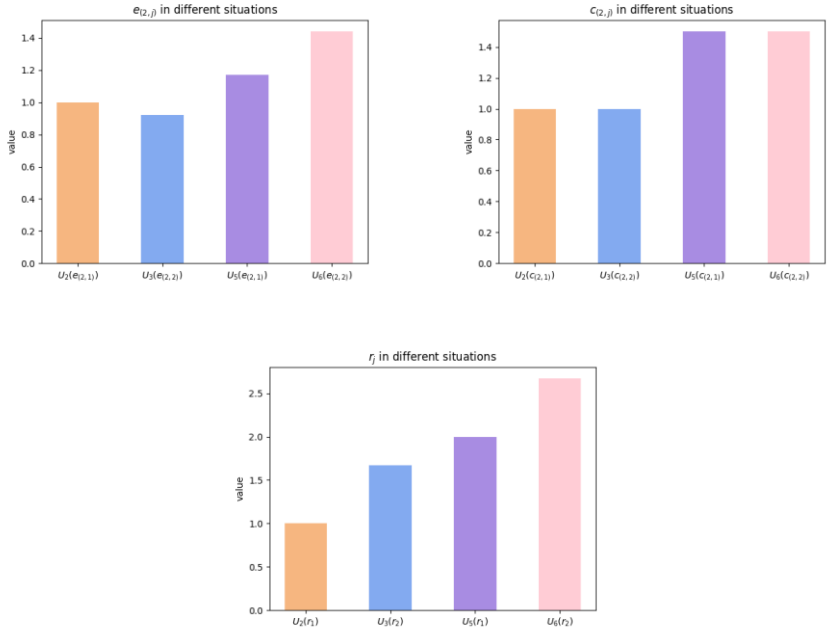

Fig. 6. Values of $e_{(2,j)}, c_{(2,j)}, r_j$ in different scenarios.

Regarding the distribution of $e_{(2,j)}$ values in Figure 6, when redirecting unknown types of malicious traffic, the benefits for the present and the future will be the greatest. This is because when this unknown type of malicious traffic attacks next time, we can identify it as a known type of malicious traffic, reduce response costs, and therefore increase future profits.

Regarding the distribution of $c_{(2,j)}$ values in Figure 6, the response costs of the two defense strategies for known types of malicious traffic are the same, and the response costs of the two defense strategies for unknown types of malicious traffic are the same, but for unknown types of malicious traffic The response cost is higher than that of known types, because unknown types of malicious traffic require multiple features when locating it, which increases the difficulty of traffic blocking and increases the response cost.

Regarding the distribution of r_j values in Figure 6, the risk level of known types of malicious traffic is less than that of unknown types of malicious traffic, and the risk level of traffic discarding is less than that of traffic redirection, because we do not understand the intention of unknown types of malicious traffic and need to pass the traffic. This can only be known after analysis after redirection, which increases the risk of the Industrial Internet of Things being compromised. At the same time, after redirecting traffic, there is a certain probability that the attacker will see through it and be unable to discover its intentions, resulting in a risk that still exists in the event of a secondary attack.

In view of the quantitative relationship between e, c, and r, we further use numerical analysis methods to calculate the income value of the static Bayesian game model. Considering the actual situation of both offense and defense in IIoT, we assign specific values to the parameters, as shown in Table 4. Although these parameters set selected values, reasonable changes in these parameter values can also obtain similar trends in experimental results.

Table 4. Numerical analysis results

Parameter	Value
$U_2(e_{(2,1)})$	90
$U_2(c_{(2,1)})$	20
$U_2(r_1)$	30
$U_3(e_{(2,2)})$	83
$U_3(c_{(2,2)})$	20
$U_3(r_2)$	50
$U_5(e_{(2,1)})$	105
$U_5(c_{(2,1)})$	30
$U_5(r_1)$	60
$U_6(e_{(2,2)})$	130
$U_6(c_{(2,2)})$	35
$U_6(r_2)$	80

Put the parameters in Table 4 into the utility matrix in Figure 3 and Formula 8 to get the utility matrix with Figure 7 and Formula 9 with specific utility.

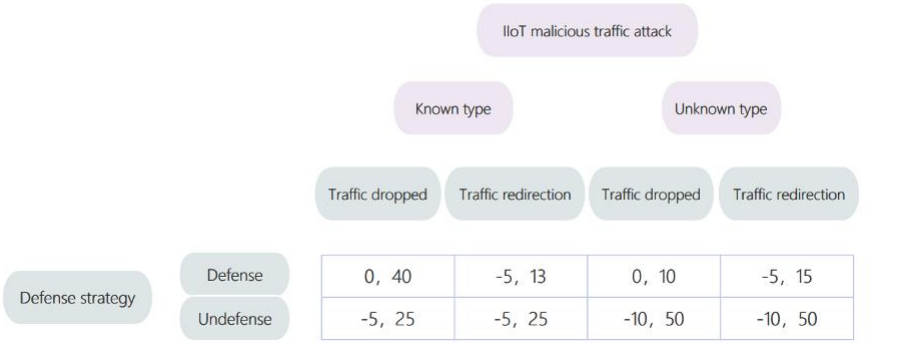

Fig. 7. Utility matrix

$$\begin{cases} b_1 = 40 \cdot p + 10 \cdot (1-p) \\ b_2 = 40 \cdot p + 15 \cdot (1-p) \\ b_3 = 13 \cdot p + 10 \cdot (1-p) \\ b_4 = 13 \cdot p + 15 \cdot (1-p) \end{cases} \tag{9}$$

Among them, $p \in [0,1]$, then according to the change of p, the change trend of $b_i (i = 1,2,3,4)$ is shown in Figure 8.

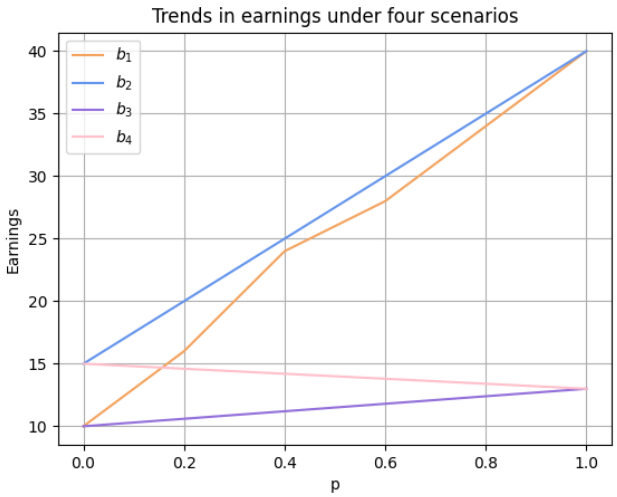

Fig. 8. The changing trend of b_i with p

As can be seen from Figure 8, when p changes from 0 to 1, the overall benefit of the scenario corresponding to b_2 is the largest, that is, traffic discarding of known types of malicious traffic and traffic redirection of unknown types of malicious traffic.

4.3 Experimental results and analysis

At the same time, we compare our method with four other recent IIoT active defense methods, which are CICDA[3], DBHM[15], and Honeypot-ML[16]. These methods are introduced in related work. We also measure the defense benefits of the above defense system using Formula 5, which includes current and future benefits (e), response cost (c), and risk level (r).

For CICDA [3], this method uses three simulated deception environments to deceive attackers. Honeynet is used to induce low-level attacks, which requires less computing, network and storage resources, so the response cost is lower and the risk level is also lower. However, the current and future benefits are also lower; Pseudocomb induces intermediate attacks, which requires more resources than Honeynet, so the response cost and risk level increase, while the current and future benefits also increase; Honeyclone induces advanced attacks, which is different from the real environment Almost

the same, so the cost of maintenance is the highest, while the benefit is the highest now and in the future. We make an overall estimate of the benefits of the CICDA method through the placement process of three honeypots.

For DBHM [15], this method uses honeypots to collect attacker information, consume attackers' resources and time for APTs, and simultaneously models dynamic attacks and defenses through prospect theory (PT). We use the modeling process and honeypots to the placement process provides an overall estimate of the benefits of the DBHM method.

For Honeypot-ML [16], this method combines honeypots with machine learning to detect botnets, classifies botnets through machine learning, and then uses honeypot detection. We use machine learning classification process and honeypot placement process to an overall estimate of the benefits of the Honeypot-ML method.

Through theoretical, numerical and experimental analysis, the income statement of the above method is shown in Table 5, and the overall income comparison is shown in Figure 9.

Table 5. Numerical analysis results of three methods

Method	Parameter	Value
CICDA	e	70
	c	19
	r	40
DBHM	e	85
	c	24
	r	40
Honeypot-ML	e	75
	c	35
	r	30

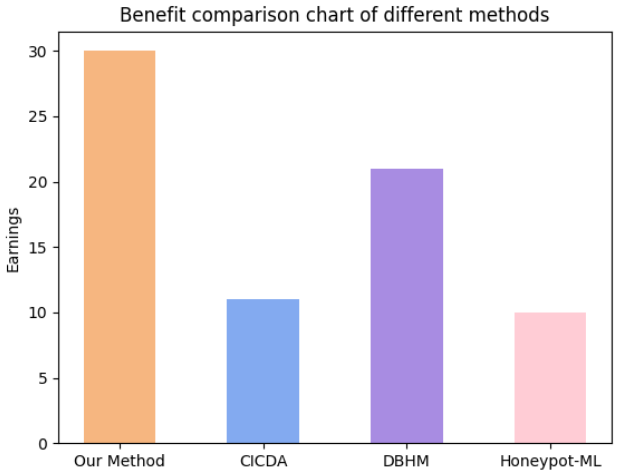

Fig. 9. Benefit comparison chart of different methods

As can be seen from Table 5 and Figure 9, our method can achieve the greatest overall benefit in solving IIoT security problems.

5 Conclusions

We propose an active defense method for IIoT security issues, which is a method of dynamically adjusting malicious traffic in the industrial Internet of Things using SDN. This traffic blocking method is divided into two types: traffic discarding and traffic redirection. For these two methods and two types of IIoT malicious traffic, we conduct attack and defense game theory analysis, and analyze them through static Bayesian game theory, numerical analysis, and experimental analysis, Find the optimal strategy for handling IIoT malicious traffic, which involves discarding traffic of known types and redirecting traffic of unknown types. At this point, the defense strategy has the greatest benefit. After comparing with other recent active defense strategies for addressing IIoT security issues, it was found that our method can achieve the maximum benefits while effectively defending.

Acknowledgments. This work was supported by the Shandong Provincial Natural Science Foundation of China 388 under Grant (No. ZR202111260301) and the Shandong Province Agricultural Major Application 389. Technology Innovation Project of China under Grant (No. SD2019NJ007).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Wang M, Zhang B, Zang X, et al. Malicious Traffic Classification via Edge Intelligence in IIoT[J]. Mathematics, 2023, 11(18): 3951.
2. Fu L, Zhang W, Tan X, et al. An algorithm for detection of traffic attribute exceptions based on cluster algorithm in industrial internet of things[J]. IEEE Access, 2021, 9: 53370-53378.
3. Neupane R L, Zobrist T, Neupane K, et al. CICADA: Cloud-based Intelligent Classification and Active Defense Approach for IoT Security[C]//IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2023: 1-6.
4. Liu Y, Tong K D, Mao F, et al. Research on digital production technology for traditional manufacturing enterprises based on industrial Internet of Things in 5G era[J]. The International Journal of Advanced Manufacturing Technology, 2020, 107(3): 1101-1114.
5. Huang H, Ye P, Hu M, et al. A multi-point collaborative DDoS defense mechanism for IIoT environment[J]. Digital Communications and Networks, 2023, 9(2): 590-601.
6. Ge M, Cho J H, Kim D, et al. Proactive defense for internet-of-things: moving target defense with cyberdeception[J]. ACM Transactions on Internet Technology (TOIT), 2021, 22(1): 1-31.
7. Bi J, He S, Luo F, et al. Defense of advanced persistent threat on industrial internet of things with lateral movement modelling[J]. IEEE Transactions on Industrial Informatics, 2022.
8. Zhang X, Wang Y, Yang M, et al. Toward concurrent video multicast orchestration for caching-assisted mobile networks[J]. IEEE Transactions on Vehicular Technology, 2021, 70(12): 13205-13220.
9. Wang Z, Jiang D, Lv Z. AI-assisted trustworthy architecture for industrial IoT based on dynamic heterogeneous redundancy[J]. IEEE Transactions on Industrial Informatics, 2022, 19(2): 2019-2027.
10. Abou El Houda Z, Brik B, Ksentini A, et al. When federated learning meets game theory: A cooperative framework to secure iiot applications on edge computing[J]. IEEE Transactions on Industrial Informatics, 2022, 18(11): 7988-7997.
11. Gan C, Lin J, Huang D W, et al. Equipment classification based differential game method for advanced persistent threats in Industrial Internet of Things[J]. Expert Systems with Applications, 2024, 236: 121255.
12. Zhang X, Wang T. Elastic and reliable bandwidth reservation based on distributed traffic monitoring and control[J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(12): 4563-4580.
13. Machado B S, Silva J M C, Lima S R, et al. Balancing the Detection of Malicious Traffic in SDN Context[C]//2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN). IEEE, 2021: 106-111.
14. Sarkar J L, Ramasamy V, Majumder A, et al. I-Health: SDN-based fog architecture for IIoT applications in healthcare[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2022.
15. Tian W, Du M, Ji X, et al. Honeypot detection strategy against advanced persistent threats in industrial internet of things: A prospect theoretic game[J]. IEEE Internet of Things Journal, 2021, 8(24): 17372-17381.
16. Lee S, Abdullah A, Jhanjhi N, et al. Classification of botnet attacks in IoT smart factory using honeypot combined with machine learning[J]. PeerJ Computer Science, 2021, 7: e350.
17. Holmgren A J, Jenelius E, Westin J. Evaluating strategies for defending electric power networks against antagonistic attacks[J]. IEEE Transactions on Power Systems, 2007, 22(1): 76-84.

18. Ismail Z, Leneutre J, Bateman D, et al. A game-theoretical model for security risk management of interdependent ict and electrical infrastructures[C]//2015 IEEE 16th International Symposium on High Assurance Systems Engineering. IEEE, 2015: 101-109.
19. Yang Y, Che B, Zeng Y, et al. MAIAD: a multistage asymmetric information attack and defense model based on evolutionary game theory[J]. Symmetry, 2019, 11(2): 215.
20. Chen X, Liu X, Zhang L, et al. Optimal defense strategy selection for spear-phishing attack based on a multistage signaling game[J]. IEEE Access, 2019, 7: 19907-19921.
21. Chen X, Feng W, Luo Y, et al. Defending against link flooding attacks in internet of things: A bayesian game approach[J]. IEEE Internet of Things Journal, 2021, 9(1): 117-128.
22. Babbar H, Rani S, AlQahtani S A. Intelligent edge load migration in SDN-IIoT for smart healthcare[J]. IEEE Transactions on Industrial Informatics, 2022, 18(11): 8058-8064.
23. Bhayo J, Jafaq R, Ahmed A, et al. A time-efficient approach toward DDoS attack detection in IoT network using SDN[J]. IEEE Internet of Things Journal, 2021, 9(5): 3612-3630.
24. Aslam M, Ye D, Tariq A, et al. Adaptive machine learning based distributed denial-of-services attacks detection and mitigation system for SDN-enabled IoT[J]. Sensors, 2022, 22(7): 2697.
25. Al Hayajneh A, Bhuiyan M Z A, McAndrew I. Improving internet of things (IoT) security with software-defined networking (SDN)[J]. Computers, 2020, 9(1): 8.
26. Shi Z, Zhou H, de Laat C, et al. A Bayesian game-enhanced auction model for federated cloud services using blockchain[J]. Future Generation Computer Systems, 2022, 136: 49-66.
27. Caligiuri M, Galizio D, Lincetto F, et al. A bayesian game of multisource energy harvesting for batteryless iot devices[C]//2022 International Conference on Electrical and Information Technology (IEIT). IEEE, 2022: 414-419.
28. Su C L. Estimating discrete-choice games of incomplete information: Simple static examples[J]. Quantitative Marketing and Economics, 2014, 12: 167-207.
29. Zandebasiri M, Filipe J A, Soosani J, et al. An incomplete information static game evaluating community-based forest management in Zagros, Iran[J]. Sustainability, 2020, 12(5): 1750.
30. Kim H. Rural Pharmacy Access and Competition: Static Games with Machine Learning[J]. Available at SSRN 4377695, 2023.
31. Zhang H, Wang J, Yu D, et al. Active defense strategy selection based on static Bayesian game[C]//Third International Conference on Cyberspace Technology (CCT 2015). IET, 2015: 1-7.
32. Wang C, Tang W, Zhao R. Static Bayesian games with finite fuzzy types and the existence of equilibrium[J]. Information Sciences, 2008, 178(24): 4688-4698.
33. Yang Z, Xiang Y, Liao K, et al. Research on security defense of coupled transportation and cyber-physical power system based on the static Bayesian game[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 24(3): 3571-3583.
34. Wei F A N, Cheng P, Dali Z H U, et al. Research on intrusion response strategy based on static Bayesian game in mobile edge computing network[J]. Journal on Communication/Tongxin Xuebao, 2023, 44(2).
35. Rahman A, Islam M J, Band S S, et al. Towards a blockchain-SDN-based secure architecture for cloud computing in smart industrial IoT[J]. Digital Communications and Networks, 2023, 9(2): 411-421.
36. Irshad A, Mallah G A, Bilal M, et al. SUSIC: A secure user access control mechanism for SDN-enabled IIoT and cyber physical systems[J]. IEEE Internet of Things Journal, 2023.
37. Jiang J, Lin C, Han G, et al. How AI-enabled SDN technologies improve the security and functionality of industrial IoT network: Architectures, enabling technologies, and opportunities[J]. Digital Communications and Networks, 2022.

38. Chandramohan S, Senthilkumaran M, Sivakumar M. Adaptive computing optimization for industrial IoT using SDN with edge computing[C]//2022 6th International Conference on Computing Methodologies and Communication (ICCMC). IEEE, 2022: 360-365.
39. Duy P T, Quyen N H, Khoa N H, et al. FedChain-Hunter: A reliable and privacy-preserving aggregation for federated threat hunting framework in SDN-based IIoT[J]. Internet of Things, 2023, 24: 100966.
40. Su J, Jiang M. A Hybrid Entropy and Blockchain Approach for Network Security Defense in SDN-Based IIoT[J]. Chinese Journal of Electronics, 2023, 32(3): 1-11.
41. Tang C, Zhu C, Zhang N, et al. Sdn-assisted mobile edge computing for collaborative computation offloading in industrial internet of things[J]. IEEE Internet of Things Journal, 2022, 9(23): 24253-24263.
42. Sharma K K, Sood M. Mininet as a container based emulator for software defined networks[J]. International Journal of Advanced Research in Computer Science and Software Engineering, 2014, 4(12).