actuators

MDPI

Article

Security State Estimation for Cyber-Physical Systems against DoS Attacks via Reinforcement Learning and Game Theory

Zengwang Jin [1,2,*], Shuting Zhang [1], Yanyan Hu [3], Yanning Zhang [2] and Changyin Sun [4]

[1] School of Cybersecurity, Northwestern Polytechnical University, Xi'an 710072, China; shutingzhang@mail.nwpu.edu.cn
[2] National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, Northwestern Polytechnical University, Xi'an 710072, China; ynzhang@nwpu.edu.cn
[3] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China; huyanyan@ustb.edu.cn
[4] School of Automation, Southeast University, Nanjing 210096, China; cysun@seu.edu.cn
* Correspondence: jin_zengwang@nwpu.edu.cn

Abstract: This paper addressed the optimal policy selection problem of attacker and sensor in cyber-physical systems (CPSs) under denial of service (DoS) attacks. Since the sensor and the attacker have opposite goals, a two-player zero-sum game is introduced to describe the game between the sensor and the attacker, and the Nash equilibrium strategies are studied to obtain the optimal actions. In order to effectively evaluate and quantify the gains, a reinforcement learning algorithm is proposed to dynamically adjust the corresponding strategies. Furthermore, security state estimation is introduced to evaluate the impact of offensive and defensive strategies on CPSs. In the algorithm, the ε-greedy policy is improved to make optimal choices based on sufficient learning, achieving a balance of exploration and exploitation. It is worth noting that the channel reliability factor is considered in order to study CPSs with multiple reasons for packet loss. The reinforcement learning algorithm is designed in two scenarios: reliable channel (that is, the reason for packet loss is only DoS attacks) and unreliable channel (the reason for packet loss is not entirely from DoS attacks). The simulation results of the two scenarios show that the proposed reinforcement learning algorithm can quickly converge to the Nash equilibrium policies of both sides, proving the availability and effectiveness of the algorithm.

Keywords: cyber-physical system; security estimation; DoS attack; reinforcement learning; Nash equilibrium

Citation: Jin, Z.; Zhang, S.; Hu, Y.; Zhang, Y.; Sun, C. Security State Estimation for Cyber-Physical Systems against DoS Attacks via Reinforcement Learning and Game Theory. *Actuators* **2022**, *11*, 192. https://doi.org/10.3390/act11070192

Academic Editor: Gang Wang

Received: 16 June 2022
Accepted: 14 July 2022
Published: 16 July 2022

1. Introduction

With the close interaction and integration between computational and physical resources, cyber-physical systems (CPSs) have emerged and gained widespread attention. Due to the application of 3C (computation, communication, control) technologies, CPSs can perform real-time sensing and remote control [1]. Therefore, CPSs are widely applied in critical infrastructure control, aerospace systems, military systems, etc. [2]. In CPSs, wireless sensors are widely used due to their flexibility, power saving and easy scalability [3]. However, while improving communication efficiency, the transmission of measurement data over wireless networks poses security issues such as significant damage to industrial systems [4]. For example, Sberbank, Russia's largest bank, was hit by the largest DDoS attack ever with peak traffic of 450 GB/s in May 2022, posing a huge threat to its cybersecurity. The frequent occurrence of such malicious cyber attacks has led many scholars and experts to pay great attention to security issues in CPSs. Hence, CPSs under cyber attacks are studied and countermeasures are proposed to ensure the security of the systems [5,6].

In terms of specific attack types, cyber attacks encountered on CPSs are mainly divided into DoS attacks [7,8], spoofing attacks [9,10] and injection attacks [11–13]. Among them,

DoS attacks have been paid more attention as one of the most frequent and easy cyber attacks to implement. They mainly prevent remote estimators from receiving and processing sensor data properly by interfering with the communication channel. The existing modeling approaches for the DoS attack problem are mainly divided into attack constraint modeling and stochastic modeling. The former is a modeling approach that constrains the duration and switches of DoS attacks. For example, Refs. [14,15] puts constraints on the frequency and duration of DoS attacks and focuses on finding the maximum attack frequency and maximum attack duration while maintaining system performance. The latter is a stochastic modeling approach related to Bernoulli distribution and Markov chain. This paper conducts a study on the latter modeling approach, where DoS attacks occur in accordance with Bernoulli distribution. Furthermore, the attack frequency is random and the attack duration is the sampling interval.

For CPSs under DoS attacks, numerous scholars have carried out research on state estimation based on systematic measurement sequences and prior models, with methods such as Kalman filtering, see [16–19] and references therein. From the attacker's perspective, most of the existing literature focuses on the DoS attack scheduling strategies. In [17], the problem of optimal scheduling of DoS attacks under energy-limited conditions was investigated based on signal-to-interference-plus-noise ratio (SINR) of the channel. Some literature, on the other hand, stand from the defender's perspective and consider the stability of the system under DoS attacks, or propose DoS attack detection methods [18–20]. Among them, a method to protect state privacy by maximizing the state estimation error of eavesdroppers for energy-constrained sensors was studied in [20]. Furthermore, the PMU data of the bus was used by Hasnat in [18] to perform state estimation of the attacked components of the system under different DoS attack strengths, so as to attenuate the impact brought by the attacks. Different from the above literature, this paper not only uses the state estimation method to evaluate the status of CPSs under DoS attacks, but also introduces game theory to study the optimal strategies of both attackers and defenders.

In CPSs, some scholars study the relationship between DoS attackers and system defenders in the system, and regard the confrontation between the two as a two-player game [21,22]. In some papers, the interactive game is further formulated as a Markov decision process, by optimizing the behavior strategy of the attacker or defender to meet the needs of its interaction with the environment [23]. The game of attackers and defenders is transformed into a static Bayesian game to obtain the optimal strategy for both sides in [24]. Furthermore, some papers considered how to reach a Nash equilibrium policy under the game between attackers and defenders, that is, neither player unilaterally changes his strategy under the Nash equilibrium to improve his own reward when the other keeps theirs unchanged [25–27]. The interactive game process between the sensor and the attacker as a Markov game framework and employs an improved Nash learning algorithm in order to obtain a Nash equilibrium for the two sides was investigated in [26]. Since the reward gained by the attacker comes entirely from the loss of the defender, ref. [28] treated the game between the attacker and the defender as a zero-sum matrix game and designs a time-difference (TD) learning-based algorithm to obtain the optimal attack strategy. Based on the above discussions, a two-player zero-sum deterministic game is introduced in this paper to describe the interactive decision process between the attacker and the sensor. Since the existing static game methods cannot fully satisfy the demands of real-time state update in CPSs, this paper adopts the linear programming method to obtain the Nash equilibrium strategy of both sides as the optimal strategy.

With the development of artificial intelligence, reinforcement learning methods have attracted much attention, focusing on how agents learn optimal strategies by interacting with unknown environments [29]. In recent research, reinforcement learning methods have been used to solve the game problem between attackers and system defenders in CPSs [30,31]. Numerous studies about reinforcement learning algorithms were carried out in different scenarios. In [32], reinforcement learning is classified into model-based and model-free approaches, and a model-free reinforcement learning method is designed to

solve the attacker's security-aware planning. Furthermore, the game between attackers and sensors from the open-loop case and closed-loop case is studied in [33], and centralized and distributed reinforcement learning methods are proposed to solve the Nash equilibrium of the two sides. Furthermore, some limitations of CPSs make it difficult to gather collective information and perform state estimation efficiently. For instance, sensor devices are often small in size and carry energy-constrained batteries, which are hard to replace in some situations. Moreover, bandwidth resources are also limited, which result in network congestion or packet loss with the increase of the number of sensor nodes. It is necessary to optimize the utilization of system resources by reducing the needless consumption of system resources. Up to now, little attention has been paid to the optimal strategy obtained by reinforcement learning from the perspective of channel reliability and resource utilization, which is also one of the motivations of this paper.

In summary, this paper investigates the problem of the remote security state estimation problem of CPSs with DoS attacks, where a reinforcement learning algorithm to achieve the Nash equilibrium policies for the two-player game between both the attacker and the sensor. This paper develops from two scenarios: reliable channel and unreliable channel, where the reason for packet loss in reliable channel transmission can only be DoS attacks, while unreliable channel transmission may lose packets due to other reasons. The contributions of the paper are as follows: (i) This paper introduces security state estimation into existing reinforcement learning methods to evaluate the impact of the policies of attackers and defenders on the state estimation; (ii) A two-player zero-sum game is introduced to describe the game between the sensor and the attacker, and the Nash equilibrium strategies are studied to obtain the optimal actions. Besides, resource constraints for the sensor and the attacker are considered in the game; (iii) Reinforcement learning algorithms are designed to enable sensors and attackers dynamically learn and adjust policies in the interaction, where ε-greedy policy is improved to achieve a balance of exploration and exploitation; (iv) Considering the influence of channel reliability on CPSs, the reinforcement learning algorithm is studied in two scenarios: reliable channels and unreliable channels, the packet loss probability of the two scenarios is compared.

The rest of this paper is organized as follows. Section 2 formulates the system model and introduced some preparatory knowledge of a two-player zero-sum game and the Q-learning algorithm. In Section 3, the state estimation algorithm based on Kalman filter is designed, and the influence of the DoS attack on state estimation is described. Reinforcement learning algorithms for reliable and unreliable channels are designed in Sections 4 and 5, respectively. The simulation results of two cases in Section 6 illustrate the effectiveness and efficiency of the reinforcement learning algorithm. Section 7 draws the conclusion and discusses the future direction.

2. Preliminaries

Section 2 provides a theoretical basis for this paper which can better explain the methods used in the paper, so that the reader better understands the content. The section formulates the system model and introduces some preparatory knowledge of the two-player zero-sum game and Q-learning algorithm. In the part of the system model, the principle of remote security estimation of a wireless channel under DoS attack is described in detail. The preparatory knowledge of the two-player zero-sum game includes the definition of the two-player zero-sum game and the definition of the pure strategy Nash equilibrium and mixed strategy Nash equilibrium. The preparatory knowledge of the Q-learning algorithm include the introduction of the Markov decision process and the iteration of the one-step Q-learning method according to the Bellman optimal equation.

2.1. System Model

Consider a CPS with a sensor, attacker and remote estimator as presented in Figure 1; a DoS attack may occur on the wireless communication channel between the sensor and

the remote state estimator. At time k, the expression of the linear system under DoS attacks can be given by:

$$\begin{cases} x(k+1) = Ax(k) + Bw(k) \\ y(k) = Hx(k) + v(k), \end{cases} \tag{1}$$

where $k \in \mathbb{Z}$ denotes the discrete time step. $x(k) \in \mathbb{R}^{d_x}$ refers to the state vector of the system, $y(k) \in \mathbb{R}^{d_y}$ is the sensor measurement vector. $w(k) \in \mathbb{R}^{d_w}$ and $v(k) \in \mathbb{R}^{d_v}$ represent the process and measurement noises with zero mean, and their covariance matrices are $Q(k)$ and $R(k)$, respectively. A, B, and H are coefficient matrices with corresponding dimensions.

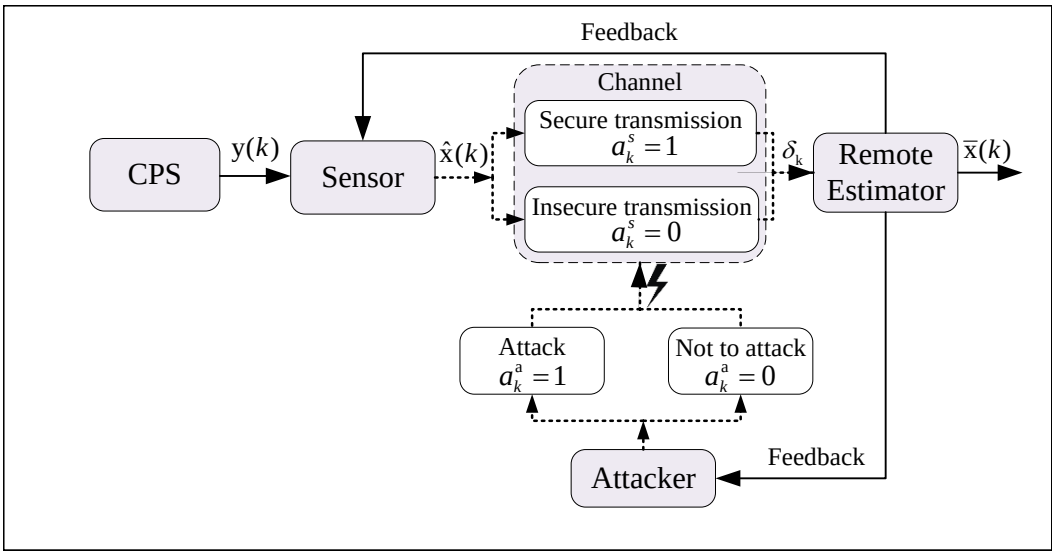

Figure 1. Remote security estimation over wireless channel under DoS attacks.

2.2. Two-Player Zero-Sum Game

In this paper, a two-player zero-sum deterministic game is considered to describe the interaction between the sensor and the attacker. The definition of the zero-sum game is as follows [34]:

Definition 1. *The basic game model that meets the following three conditions is called the zero-sum matrix game: (i) The number of players in the game is two, namely* $N = \{n_1, n_2\}$*; (ii) Each player's strategy space is a finite set. That is, the strategy spaces* $\Pi_1 = \{\alpha_1, \alpha_2, \cdots, \alpha_m\}$ *and* $\Pi_2 = \{\beta_1, \beta_2, \cdots, \beta_n\}$ *are finite sets, and any strategy* α_i *and* β_j *can be taken out from the strategy spaces* Π_1 *and* Π_2 *to form a strategy combination* (α_i, β_j)*; (iii) For any strategy combination* (α_i, β_j)*, the payoffs of the players are* $U_1(\alpha_i, \beta_j)$ *and* $U_2(\alpha_i, \beta_j)$ *respectively, and* $U_1(\alpha_i, \beta_j) + U_2(\alpha_i, \beta_j) = 0$*.*

In CPSs under attacks, there are only two players in the game: the sensor and the attacker. The two players choose their strategies in a limited set of policies, that is, the sensor can choose a secure or insecure transmission channel and the attacker can choose to attack or not. The goal of the sensor is to minimize the state estimation error and maximize the attack cost, while the attacker has the opposite goal. Since the basic conditions of a zero-sum game are satisfied, and there is no randomness in the action, the game can be formulated as a zero-sum deterministic game.

The Nash equilibrium policy of the zero-sum deterministic game is a joint policy, in which each player considers the behavior of the other player and makes the best response. Nash equilibrium is divided into pure strategy Nash equilibrium and mixed strategy Nash equilibrium, which are defined as follows.

Definition 2. *For the two-player zero-sum game, a pure strategy Nash equilibrium is a joint policy that each player sticking to only one choice that is most beneficial to him, which is a combination of actions* (a_*^1, a_*^2) *such that:*

$$\begin{array}{ll} r^1(a_*^1, a_*^2) \geq r^1(a^1, a_*^2) & \text{for all } a^1 \in A^1 \\ r^2(a_*^1, a_*^2) \geq r^2(a_*^1, a^2) & \text{for all } a^2 \in A^2, \end{array}$$

where r^1 *and* r^2 *correspond to the rewards of the two players, respectively, and* A^1 *and* A^2 *are the action sets.*

Definition 3. *A mixed strategy Nash equilibrium of two-player zero-sum game allow players to choose actions probabilistically to give the best response to the action of the other player. It is an action combination represented by a vector* (ω_*^1, ω_*^2) *such that:*

$$\begin{array}{ll} \omega_*^1 R^1 \omega_*^2 \geq \omega^1 R^1 \omega_*^2 & \text{for all } \omega^1 \in \Omega(A^1) \\ \omega_*^1 R^2 \omega_*^2 \geq \omega_*^1 R^2 \omega^2 & \text{for all } \omega^2 \in \Omega(A^2), \end{array}$$

where ω_*^1 *and* ω_*^2 *are non-negative vectors whose entries sum to 1,* $\Omega\left(A^k\right)$ *is the set of probability distributions over the action space* A^k*,* R^1 *and* R^2 *are the payoff matrices of two players.*

In 1950, Nash proved that there is at least one Nash equilibrium in any finite time-horizon game [35]. Therefore, we assume that there exist Nash equilibrium policies $\{\pi_1^*, \pi_2^*\}$ for all state $s \in S$, where π_1^* is the policy for sensor and π_2^* is the policy for the attacker.

2.3. Q-Learning Algorithm

As an important branch of artificial intelligence, reinforcement learning is a mathematical framework that helps learners to interact with an unknown environment in order to achieve their goals. Markov decision process (MDP) is one of the theoretical bases of reinforcement learning, which refers to the sequential decision-making problem in completely observable random environment with the Markov transition model and rewards [36]. The Markov decision process consists of five tuples, which are defined as follows: $MDP ::= \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \rho \rangle$ [37]. In this defining equation, $\mathcal{S}$ and $\mathcal{A}$ represent the discrete state space and discrete action space, respectively. The agent's reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ indicates the reward that an agent can get by choosing an action in a certain state. The state transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to [0, 1]$ denotes the probability that the agent chooses an action and enters the next state. ρ is the discount factor between 0 and 1.

Q-learning is an essential reinforcement learning approach, which is an off policy learning algorithm based on the Markov decision process. It was put forward by Watkins [38] in 1989, and is currently a widely used reinforcement learning approach. The Q-learning algorithm enables the agent to select the optimal strategy and maximize the reward by evaluating the current system state and actions without considering the external environment model.

Assume that both the set of state S and the set of action A are finite sets. Then the optimal action cost function $Q(s, a)$ can be expressed as a table with m rows and n columns, where m is the number of states and n is the number of actions. While the system is in state s_k, the decision formula of time k is:

$$a_k = \underset{a \in A}{\operatorname{argmax}} Q_\star(s_k, a). \tag{2}$$

According to the Bellman optimal equation in [39], the iteration of the one-step Q-learning method is derived as:

$$\begin{aligned} Q(s_k, a_k) \leftarrow Q(s_k, a_k) + \alpha[r_{k+1} \\ + \rho \max_a Q(s_{k+1}, a) - Q(s_k, a_k)], \end{aligned} \tag{3}$$

where α is the learning rate, Q-value function is the learned state-action value function.

3. Problem Statement

3.1. State Estimation Based on Kalman Filter

A local Kalman filter is employed to complete the state estimation of recursive updating of the system state. At each time k, the minimum mean-squared error (MMSE) estimate $\hat{x}(k)$ of the state vector $x(k)$ is obtained according to the measurement data by running the Kalman filter. Then, $\hat{x}(k)$ is sent to the remote estimator by the sensor. The MMSE estimate $\hat{x}(k)$ of $x(k)$ is denoted by:

$$\hat{x}(k) = \mathbf{E}[x(k) \mid y(1), \ldots, y(k)] \tag{4}$$

with its corresponding estimation error covariance

$$\begin{aligned} P(k) = \mathbf{E}\Big[(x(k) - \hat{x}(k))(x(k) - \hat{x}(k))^T \\ \mid y(1), \ldots, y(k)]. \end{aligned} \tag{5}$$

In accordance with the Kalman filter equations, $\hat{x}(k)$ and $P(k)$ are updated recursively. For simplicity, the Lyapunov and Riccati operators h and $\tilde{g}$ is defined as:

$$\begin{aligned} h(X) &\triangleq AXA^T + Q \\ \tilde{g}(X) &\triangleq X - XC^T\left[CXC^T + R\right]^{-1}CX. \end{aligned} \tag{6}$$

Then the recursive updating equation of the Kalman filter can be expressed as:

$$\begin{aligned} &\hat{x}(k \mid k-1) = A\hat{x}(k-1) \\ &P(k \mid k-1) = h(P(k-1)) \\ &K(k) = P(k \mid k-1)C^T\left[CP(k \mid k-1)C^T + R\right]^{-1} \\ &\hat{x}(k) = \hat{x}(k \mid k-1) + K(k)(y(k) - C\hat{x}(k \mid k-1)) \\ &P(k) = g(P(k \mid k-1)). \end{aligned} \tag{7}$$

Due to the stabilizability and detectability assumptions, the estimation error covariance $P(k)$ converges to a unique fixed point $\bar{P}(k)$ of $h \circ g$ at an exponential rate [40]. In order to simplify the subsequent reinforcement learning problem, we assume that the Kalman filter has already converged to the steady state, i.e., for all $k \geq 1, P(k) = \bar{P}$.

3.2. DoS Attack Model

According to the system model shown in Figure 1, a DoS attack can occur in the channel between the sensor and the remote estimator. The likelihood of DoS attacks varies depending on the choice of the sensor and the attacker. For the sensor, there are two choices of sending data, which can be denoted as $a^s = 0$ and $a^s = 1$. The former notation means that the sensor sends packets at a cost of 0, but is vulnerable to DoS attacks; the latter notation means that the sensor spends extra cost c_s to avoid attacks. On the other hand, the DoS attacker can choose to attack or not and the attack will induce additional constant cost c_a.

Denote $a_k^s \in \{0, 1\}$ and $a_k^a \in \{0, 1\}$ as the decision variable of the sensor and the attacker at time k. The sensor are more prone to DoS attacks when $a_k^s = 0$, while it transmits in a safer way when $a_k^s = 1$. In addition, the DoS attacker chooses to attack at time k when $a_k^a = 1$, and no attack happens when $a_k^a = 0$.

In CPSs, packet loss may occur due to DoS attacks or other reasons such as channel congestion. To indicate the arrival of packets and to determine whether packet loss has occurred at time k, we define the arrival indicator δ_k as:

$$\delta_k = \begin{cases} 1, & \text{if packet arrives at the remote estimator} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Furthermore, the packet loss probability $\xi_k \in (0, 1)$ is defined to describe the proportion of packets lost in data transmission to the total packets sent. When the action combination of the sensor and the attacker is easy to induce attacks, the overall performance of the network will decline, and the packet loss probability ξ_k will be high.

Packet loss is not only related to the choice of sensors and attackers, but also to the reliability of the communication channel. We will illustrate the impact of channel reliability on the packet loss probability by showing the occurrence of attacks under reliable and unreliable channels in Sections 4 and 5, respectively.

Remark 1. *In this paper, DoS attacks are modeled as stochastic dropouts, and the filtering process is similar to that of data loss. However, there are some differences between the two processes. Similar to transmission delay, channel fading, etc., data loss and intermittent observation are intrinsic phenomena in the network. They are passive representations of network unreliability and instability. On the contrary, this paper pays more attention to the game of the attackers and defenders and their Nash equilibrium points, which is a dynamic game process of active attack and active defense. From passive description to active attack confrontation, this paper combines game theory and reinforcement learning methods to evaluate and dynamically adjust the dynamic game process between attackers and defenders.*

3.3. Remote Estimation

At the remote estimator, the estimation process of $\hat{x}(k)$ can be described as follows: if the estimator receives the packet successfully, it synchronizes the received data packet to obtain the estimation. Otherwise, the estimator updates the estimation based on the optimal estimation obtained at the previous time step, i.e.,

$$\bar{x}(k) = \begin{cases} \hat{x}(k), & \delta_k = 1 \\ A\bar{x}(k-1), & \text{otherwise,} \end{cases} \tag{9}$$

with the corresponding estimation error covariance

$$P(k) = \begin{cases} \bar{P}, & \delta_k = 1 \\ h(P(k-1)), & \text{otherwise.} \end{cases} \tag{10}$$

In order to simplify the error covariance $P(k)$, the indicator τ is defined as:

$$\tau_k \triangleq k - \max_{0 \le l \le k} \{l : \delta_l = 1\}, \tag{11}$$

which indicates the time interval from the last time l when packet was received to the current time k. The iteration of τ_k can be expressed as:

$$\tau_k = \begin{cases} 0, & \text{if } \delta_k = 1 \\ \tau_{k-1} + 1, & \text{otherwise.} \end{cases} \tag{12}$$

We make the following assumption that at the beginning of the transmission, the estimator can receive the packet $\hat{x}_0$ successfully, that is, $\delta_0 = 1$. Based on (10) and (11), it is easy to obtain the estimation error covariance at the remote estimator as $P(k) = h^{\tau_k}(\bar{P})$.

In the game between sensors and attackers, there are differences in the information they can obtain. The sensor knows the system model parameters, so $P(k)$ is available at time k. Therefore, the sensor's action selection is based on the following cost minimiza-

tion principle: when the safe transmission cost c_s is less than the cost caused by packet loss, i.e., $c_s \leq Tr(P(k)) - Tr(\bar{P})$, the sensor can only choose $a_k^s = 1$. However, for the attacker, the system model parameters are not available, so there is no principle for its action selection.

In accordance with the above principle, the set of values of τ_k and $P(k)$ are finite sets. Due to τ_k, take values from $\mathbb{Z}_k \triangleq \{0, 1, 2, \ldots, k\}$, $P(k)$ take values from $\mathbb{P} \triangleq \{\bar{P}, h(\bar{P}), h^2(\bar{P}), \ldots, h^n(\bar{P}), \}$. Notice that the current state $P(k)$ depends only on the last state $P(k-1)$ and δ from (8). Hence, the sequence of random states $P(k)$ forms a Markov chain, and the transition process is depicted in Figure 2. With the actions of the two agents a_k^s and a_k^a, the transition can be described by a simple transition probability matrix

$$T = \begin{pmatrix} 1-\xi_0 & \xi_0 & & & \\ 1-\xi_1 & & \xi_1 & & \\ 1-\xi_2 & & & \xi_2 & \\ \vdots & & & & \ddots \end{pmatrix},$$

where the element $T(i, j)$ represents the transition probability from state $s_k = h^{i-1}(\bar{P})$ to state $s_{k+1} = h^{j-1}(\bar{P})$. Based on Markov chains in Figure 2, states can only be transferred from $h^i(\bar{P})$ to the next adjacent state $h^{i+1}(\bar{P})$ or the initial state $\bar{P}$). Hence, except for all $T(i, 1)$ and $T(i, i+1)$, the other default elements are 0.

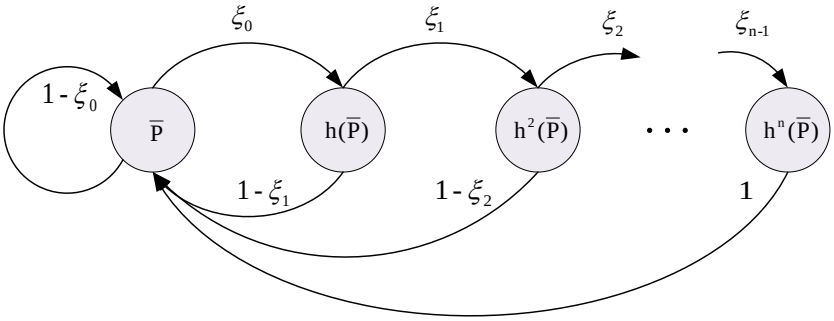

Figure 2. Markov chain transition process of P(k).

4. Reinforcement Learning for Reliable Channel

In CPSs, reliability is an important indicator of system performance [41]. The measurement data from the sensor are transmitted over a wireless channel, and packets transmitted over a reliable channel will not be corrupted or lost by timeout, and vice versa for an unreliable channel. Assuming an ideal state in which the packets are transmitted in a reliable channel, there is no congestion and timeout, and packets are lost only because of DoS attacks. According to the description of Section 3.2, the packet loss probability for a reliable channel can be expressed as:

$$\xi_k = \begin{cases} 100\%, & \text{if } a_k^s = 0, a_k^a = 1 \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

That is, when the sensor chooses to transmit insecurely and the attacker chooses to perform a DoS attack, the packet loss probability is 100% and in other cases the packet loss probability is 0.

According to the above assumptions and the existing reinforcement learning framework, an MDP is established to describe the interactive decision-making problem between sensor and attacker. The elements of MDP can be described as follows:

State: We denote the finite set of states in the reinforcement learning problem by $S \triangleq \{s_1, s_2, \cdots, s_n\}$. The state of the system is represented by the estimation error covariance of the estimator, i.e., $s_k = P(k)$, where $P(k) \in \mathbb{P}$. At time k, the state of the system is affected by the state s_{k-1} and actions a_{k-1}^s and a_{k-1}^a of sensor and attacker at time $k-1$.

Action: In the zero-sum deterministic game, the sensor needs to choose whether to spend cost c_s for secure transmission. Besides, the DoS attacker will choose to implement the DoS attack or not. It should be noted that the action selection of sensor and attacker are independent, that is to say, the actions of one player will not affect the other player. According to the description in the DoS attack model, the decision variable of the sensor is $a_k^s \in \{0,1\}$ and the decision variable of the attacker is $a_k^a \in \{0,1\}$. Hence, the sensor and the attacker both have two strategies at state s_k. There are four combinations of sensor and attacker actions at each time k. Thus, we denote A as the set of actions of $a_k = (a_k^s, a_k^a)$ and A has four elements.

State transition: As mentioned above, the estimation of the remote estimator is closely related to whether the packet loss occurs, and the same is true for the state transition. In reliable channel, DoS attack occurs when and only when the sensor chooses the insecure transmission and the attacker launches DoS attack i.e., $a_k^s = 0, a_k^a = 1$, resulting in packet loss. Hence, the state of time $k+1$ is determined by the action combination of the sensor and the attacker, and can be obtained according to (10) as follows:

$$s_{k+1} = P(k+1) = \begin{cases} h(P(k)), & \text{if } a_k^s = 0, a_k^a = 1 \\ \bar{P}, & \text{otherwise.} \end{cases} \tag{14}$$

Reward function: A reward function is defined to evaluate the payoff of both the sensor and the attacker during the game. Under the cost setting in Section 3.2, the reward of the system depends on the state of the system, as well as the cost and strategy of the attacker and the sensor. The sensor's goal is to minimize the reward function and the attacker's is the opposite. At time k, the immediate reward r_k can be calculated as:

$$r_k = Tr(s_k) + c_s a_k^s - c_a a_k^a. \tag{15}$$

Discount factor: In the reinforcement learning problem, since we prefer to focus more on the current reward rather than the future reward, a discount factor is set to reduce the impact of future rewards on the current state. The discount factor is a parameter between 0 and 1, with a time-based penalty to achieve better performance of the algorithm. By setting the discount factor, the farther the future is, the greater the discount is given to the reward, which makes the algorithm converge faster.

For the two-player zero-sum game between the sensor and the attacker, this paper proposes a Q-learning-based game algorithm to find the Nash equilibrium policies. The algorithm is divided into the following steps. Firstly, the states, actions and the game matrix based on Q values are initialized. Secondly, at each time k, the sensor and the attacker choose actions according to the game matrix by employing the ε-greedy strategy. Thirdly, the current reward as well as the next state s_{k+1} are obtained based on the current state s_k and the combination of actions $a_k = (a_k^s, a_k^a)$. Then, the Q-value matrix is adjusted and then the game process is carried out for the next moment. Finally, the converged Q matrix is obtained and the Nash equilibrium between the sensor and the attacker is observed. The algorithm can be described explicitly as follows:

Step 1: Input the system parameters, initialize the system state, action and Q-value matrix. Under the principles set in Section 3.3, given the cost and error covariance matrix, then the system can be determined to have n states, assuming that the initial state $s_1 = \bar{P}$. Since the system has n states, and each state has four combinations of sensor and attacker actions, so the Q-value matrix with n rows and four columns is initialized. The initial value of the Q-value matrix is set to m, where m satisfies $m \geq \max_{k\geq 1}\{r_k\}/(1-\rho)$, then the monotonically non-increasing property of $\widetilde{Q}(s, a^s, a^a)$ is guaranteed.

Step 2: At each time k, the sensor and the attacker select the action with the $\varepsilon - greedy$ strategy. According to this strategy, the sensor selects the action randomly with a probability of ε and the optimal action with a probability of $1-\varepsilon$.

Remark 2. *At the beginning of the iteration, the value of ε is set to be large, that is, the action selection has great randomness. As the algorithm iterates, the value of ε decreases gradually until the set minimum value is reached. The core idea of this method is to strike a balance between exploration and exploitation. Setting a large ε at the beginning of the iteration allows sensors and attackers to choose actions relatively randomly to learn the rewards for each action combination, which is called exploration. At the end of the iteration, sensors and attackers have observed some data, and the value of ε is very small, so they can choose the action that obtains the highest reward based on the existing data, which is exploitation. In this way, better actions are selected in the case of sufficient data collection, achieving a balance between exploration and exploitation.*

The optimal action is obtained by calculating the Nash equilibrium using the linear programming method in Lemma 1.

Lemma 1. *Let the value of the matrix game be $v > 0$. The optimal strategy of the sensor and the attacker is equivalent to the linear programming problem as follows:*

$$\begin{aligned}
&\max Z = v \\
&s.t. \quad \textstyle\sum_i a_{ij} x_i \geq v, j = 1, 2, \cdots, n \\
&\qquad \textstyle\sum_i x_i = 1 \\
&\qquad x_i \geq 0, i = 1, 2, \cdots, m \\
&\min Z = v \\
&s.t. \quad \textstyle\sum_j a_{ij} y_j \leq v, i = 1, 2, \cdots, m \\
&\qquad \textstyle\sum_j y_j = 1 \\
&\qquad y_j \geq 0, j = 1, 2, \cdots, n.
\end{aligned}$$

The probability distribution of the optimal strategy can be obtained by solving the linear programming problem.

According to the solution of the linear programming, the Nash equilibrium policies of the sensors and attackers can be obtained, and the two players implement their optimal actions depending on this equilibrium respectively.

Step 3: With the current state s_k and the combination of actions $a_k = (a_k^s, a_k^a)$ of sensor and attacker, the reward r_k can be calculated by (15). Meanwhile, the next state is obtained according to (14). Note that when $s_k = h^n(\bar{P})$, in accordance with the Markov chain shown in Figure 2, the next state is determined, namely $s_{k+1} = \bar{P}$.

Step 4: We use $\widetilde{Q}(s, a^s, a^a)$ to denote the Q-value function under the state s_k and action $a_k = (a_k^s, a_k^a)$. In order to update the Q-value matrix, the Q-value function is calculated according to the following iteration rules:

$$\begin{aligned}
\widetilde{Q}_{k+1}(s, a^s, a^a) &= (1 - \alpha_k)\tilde{Q}_k(s, a^s, a^a) \\
&+ \alpha_k \left(r_k + \rho \max_{a_{k+1}^a} \min_{a_{k+1}^s} \widetilde{Q}_k(s, a^s, a^a) \right),
\end{aligned} \tag{16}$$

where α is the learning rate in $(0, 1)$, which determines the extent of learning the results of new attempts. ρ is the discount factor, r_k is the immediate reward, and the Q-value matrix is obtained by the maxmin operation.

Step 5: Determine whether the termination condition is satisfied. If the termination condition is met, the algorithm terminates; otherwise k = k + 1, and go back to step 2.

Step 6: After the loop terminates, the converged Q-value matrix $\widetilde{Q}^*(s, a^s, a^a)$ is obtained and the optimal policy based on Nash equilibrium π_1^* and π_2^* can be obtained for each state, where π_1^* is the optimal policy of the sensor and π_2^* is the optimal policy of the attacker.

The Q-learning algorithm for reliable channel is presented in Algorithm 1.

Algorithm 1 Q-learning Algorithm for Reliable channel

Input : The parameters of the system A, C; the steady-state error covariance $\bar{P}$; cost c_s and c_a; learning rate α, discount factor ρ and exploration rate ε.
Output : Optimal Q-value matrix $\widetilde{Q}^*(s, a^s, a^a)$, Nash equilibrium π_1^* and π_2^*.
Initialize: Set initial state $s_1 = 0$, initialize Q-value matrix with m for all s and $a = (a^s, a^a)$, set $k = 1$.

while $\left\| \widetilde{Q}_{k+1}(s,a^s,a^a) - \widetilde{Q}_k(s,a^s,a^a) \right\| < \eta$ **do**
  if $rand < \varepsilon$ **then**
    Choose actions randomly;
  else
    Find the optimal actions obtained by linear programming method.
  end if
  Observe the reward r_k by (15).
  Observe the next state s_{k+1} according to (14).
  Update the Q-value matrix by (16).
  $s_k \longleftarrow s_{k+1}$
  $k \longleftarrow k+1$
end while
Return Q-value matrix for $\widetilde{Q}^*(s, a^s, a^a)$.
Observe the Nash equilibrium π_1^* and π_2^*.

5. Reinforcement Learning for Unreliable Channel

In practical CPSs, the channels over which the packets are transmitted is usually unreliable channels. In this scenario, packet loss can occur due to different reasons besides DoS attacks, including signal degradation, channel fading and channel congestion. Whereas the occurrence of packet loss is related to the choice of sensors and attackers, the packet loss probability of an unreliable channel can be described as follows:

$$\xi_k = \begin{cases} p_1, & \text{if } a_k^s = 0, a_k^a = 0 \\ p_2, & \text{if } a_k^s = 0, a_k^a = 1 \\ p_3, & \text{if } a_k^s = 1, a_k^a = 0 \\ p_4, & \text{if } a_k^s = 1, a_k^a = 1. \end{cases} \tag{17}$$

That is, when the sensor chooses insecure transmission and the attacker chooses not to attack, i.e., $a_k^s = 0, a_k^a = 0$, packets may be lost due to other reasons such as channel congestion, so the packet loss probability is p_1. Similarly, the packet loss probability under other action combinations can be obtained.

Remark 3. *In practical CPSs, there usually exists a relationship that $p_2 > p_4 > p_1 > p_3$, the main reasons are as follows. Firstly, the sensor choosing insecure transmission namely $a_k^s = 0$ and the attacker choosing to attack namely $a_k^a = 1$, will both cause channel insecurity and increase the packet loss probability. Secondly, DoS attacks are the main cause of packet loss, thus their impact on the packet loss probability is greater than other causes such as channel congestion.*

An MDP is set up to depict the interactive process for sensor and attacker under the framework of an unreliable channel. In the five tuples $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \rho \rangle$ of MDP, the state, action and discount factor are the same as in the reliable channel; however, the state transition and reward function are different, which can be described as follows.

State transition: When data packets are transmitted in an unreliable channel, the state transition is not only based on whether a DoS attack occurs, but also on the packet loss probability. Under the action combination of the sensor and the attacker, the packet loss probability ξ_k can be obtained according to (17). The data packet is lost with a probability of ξ_k and is not lost with a probability of $1 - \xi_k$, the corresponding arrival indicators are

$\delta_k = 0$ and $\delta_k = 1$ respectively. Hence, the state transition occurs accordingly, which can be described as:

$$s_{k+1} = P(k+1) = \begin{cases} h(P(k)), & \delta_k = 0 \\ \bar{P}, & \delta_k = 1. \end{cases} \tag{18}$$

Reward function: Since the packet loss probability changes, the reward of the system at state k has to change accordingly. If s_k and a_k are state and action, then the immediate reward r_k at time k can be obtained as:

$$r_k = Tr(\mathbf{E}[P(k)]) + c_s * a_k^s - c_a * a_k^a, \tag{19}$$

where $\mathbf{E}[P(k)]$ represents the average expectation of the remote estimation error covariance $P(k)$, which is obtained by:

$$\mathbf{E}[P(k+1)] = (1 - \xi_k) * \bar{P} + \xi_k * h(P(k)). \tag{20}$$

Remark 4. *The algorithm for unreliable channels has the following differences from the algorithm for reliable channels. First of all, the packet loss probability ξ under different combinations of actions needs to be entered first in the algorithm for unreliable channels. Second, at each time k, after the sensor and the attacker select actions according to the $\varepsilon-$ greedy strategy, an additional step is needed to obtain the packet loss probability based on the action combinations. Then, in the calculation of the reward, it is necessary to use the reward that combines the error covariance expectation in Equation (19). Finally, in the observation of the next state, Equation (18) is also used to obtain s_{k+1}.*

The Q-learning algorithm for the unreliable channel is presented in Algorithm 2.

Algorithm 2 Q-learning Algorithm for Unreliable channel

Input : The parameters of the system A, C; the steady-state error covariance $\bar{P}$; cost c_s and c_a; packet loss probability ξ_k in each action combination; learning rate α, discount factor ρ and exploration rate ε.
Output : Optimal Q-value matrix $\widetilde{Q}^*(s, a^s, a^a)$, Nash equilibrium π_1^* and π_2^*.
Initialize: Set initial state $s_1 = 0$, initialize Q-value matrix with m for all s and $a = (a^s, a^a)$, set $k = 1$.

while $\left\| \widetilde{Q}_{k+1}(s, a^s, a^a) - \widetilde{Q}_k(s, a^s, a^a) \right\| < \eta$ **do**
  if *rand* $< \varepsilon$ **then**
    Choose actions randomly;
  else
    Find the optimal actions obtained by linear programming method.
  end if
  According to the actions of sensors and attackers a_k^s, a_k^a, the packet loss probability ξ_k is obtained by (17).
  Observe the reward r_k by (19).
  Observe the next state s_{k+1} according to (18).
  Update the Q-value matrix by (16).
  $s_k \longleftarrow s_{k+1}$
  $k \longleftarrow k + 1$
end while
Return Q-value matrix for $\widetilde{Q}^*(s, a^s, a^a)$.
Observe the Nash equilibrium π_1^* and π_2^*.

Remark 5. *Whether the proposed algorithm is suitable for extending to DDoS attacks is also investigated. DDoS attacks are distributed denial of service attacks, which combines multiple computers as an attack platform to achieve the purpose of hindering the normal service of the computer or network. The attack–defense game problem under DDoS attacks adds a many-to-one dimension to the problem under DoS attacks. That is to say, DDoS attacks combine multiple attack*

sources to simultaneously attack a single sensor, and the attack cost and attack intensity of different attack sources may be different. For attackers, multiple attack sources need to be coordinated to minimize attack costs, while for system defenders, the impact of multi-source attacks needs to be minimized. In the future, we will focus on the impact of DDoS attacks on the attack and defense decisions of single-sensor systems and multi-sensor systems. One of our future directions is to coordinate multiple attack sources and sensors so that both attackers and defenders can make optimal decisions.

6. Simulations and Experiments

Two examples are given in this section to illustrate the effectiveness of reinforcement learning algorithms for reliable and unreliable channels, respectively.

6.1. Case 1: Simulation Example for Reliable Channel

Consider a CPS with parameters of system model shown as follows.

$$A = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T,$$
$$Q = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.8 \end{bmatrix}, R = 0.8.$$

The Kalman filter running converges to a steady state, and the steady state error covariance is obtained as:

$$\bar{P} = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 2.4 \end{bmatrix},$$

where trace $Tr(\bar{P})$ is 3.

The cost for the sensor to choose secure transmission is set to $c_s = 6.7$ and the attacker's attack cost is set to $c_a = 8$. In the game between the sensor and the attacker, the sensor knows the parameters of the system, so it can choose whether to adopt the costly secure transmission according to the situation. Based on the principle set in Section 3.3, the sensor can only choose secure transmission when its cost is less than the cost of continued packet loss. Due to $Tr(h^2(\bar{P})) < c_s + Tr(\bar{P}) < Tr(h^3(\bar{P}))$, it is easily learned that the sensor can only choose secure transmission after two consecutive moments of packet loss, namely $s_k = Tr(h^2(\bar{P}))$. Therefore, the system state is a finite set of $S = \{\bar{P}, h(\bar{P}), h^2(\bar{P})\}$. Due to the system settings, when $s_k = Tr(h^2(\bar{P}))$, it is bound to return to $s_k = \bar{P}$, which provides additional security for the system.

In this case, we set the learning rate and discount factor as 0.9 and 0.6, respectively. The Kalman filter as well as the Q-learning algorithm are run for 5000 iterations. During the first 500 iterations, the value of ε is set to slowly decrease from 1 to fully explore the reward for each action combination. In the subsequent iterations, ε is set to 0.1 unchanged, that is, the optimal action is selected with a probability of 0.9, and an action is randomly selected with a probability of 0.1. The optimal actions are obtained by the Nash equilibrium with the linear programming method.

The variation of $Tr(P(k))$ is plotted, as shown in Figure 3. From this figure, we can observe the occurrence of packet loss and the change of state during the learning process, where $Tr(P(k)) = 3$, $Tr(P(k)) = 5.6$ and $Tr(P(k)) = 9.6$ represent $s_k = \bar{P}$, $s_k = h(\bar{P})$ and $s_k = h^2(\bar{P})$, respectively. In the first 500 iterations, the selection of actions is random, so the frequency of packet loss is high. In the subsequent iterations, the sensor and the attacker select the optimal actions for the current situation, and it can be shown that the maximum number of consecutive packet losses is 2.

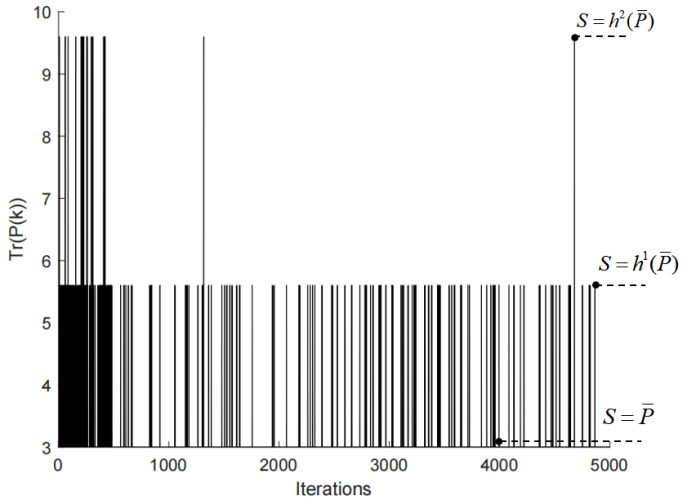

Figure 3. Variation of $Tr(P(k))$ for reliable channel.

The learning process of $\widetilde{Q}(s, a^s, a^a)$ at state $s = \{\bar{P}, h(\bar{P}), h^2(\bar{P})\}$ is shown in Figure 4. According to Figure 4, it can be concluded that the elements of the Q-value matrix decrease and converge to $\widetilde{Q}^*(s, a^s, a^a)$ by the Q-learning algorithm, and the Q-value matrix obtained by convergence is shown in Table 1. $s = \{\bar{P}, h(\bar{P}), h^2(\bar{P})\}$ represents three different states. $a = (a_k^s, a_k^a) = \{(0,0), (0,1), (1,0), (1,1)\}$ represents different action-pairs of the sensor and attacker, which has been defined in Section 3.2 and the MDPs in Section 4. The numeric values in each grid represent the convergence of $\widetilde{Q}(s, a^s, a^a)$ for each state-action.

Table 1. $\widetilde{Q}^*(s, a^s, a^a)$ for reliable channel.

State-Action	$a = (0,0)$	$a = (0,1)$	$a = (1,0)$	$a = (1,1)$
$s = \bar{P}$	7.500	2.100	14.300	6.300
$s = h(\bar{P})$	7.500	10.190	14.300	6.300
$s = h^2(\bar{P})$	7.538	−0.499	14.301	7.237

According to Nash equilibrium theory, Nash equilibrium points exist for any finite game. From the Q-value matrix, we know that there exist Nash equilibrium points of the game, which are shown in Table 2. The solution is conducted with the linear programming method to obtain the mixed strategy Nash equilibrium of the sensor and the attacker, denoted by π_1^* and π_2^*, respectively. For example, at state $s = h(\bar{P})$, the mixed strategy Nash equilibria for the sensor and the attacker are (0.748,0.252) and (0.364,0.636), respectively. This implies that the sensor chooses insecure transmission with probability 0.748 and secure transmission with probability 0.252; meanwhile, the attacker chooses no attack with probability 0.364 and attack with probability 0.636.

Table 2. Nash equilibrium π_1^* and π_2^* for reliable channel.

State-π^*	π_1^*	π_2^*
$s = \bar{P}$	$(1,0)$	$(1,0)$
$s = h(\bar{P})$	$(0.748, 0.252)$	$(0.364, 0.636)$
$s = h^2(\bar{P})$	$(1,0)$	$(1,0)$

$\pi_1^* = (Pr(a^s = 0), Pr(a^s = 1))$, $\pi_2^* = (Pr(a^a = 0), Pr(a^a = 1))$.

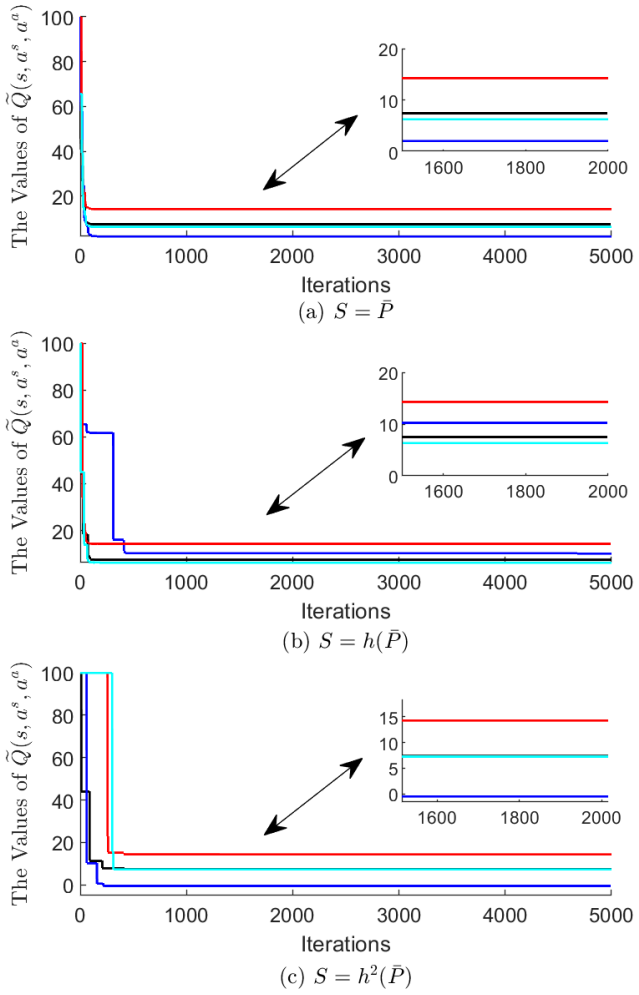

Figure 4. Learning process of $Q(s, a^s, a^a)$ for reliable channel.

After obtaining the optimal policy for the sensor and the attacker under the DoS attack of the reliable channel, we can calculate the probability of packet loss under a single attack and two consecutive attacks as an evaluation criterion for the algorithm. It can be known that the probability of two consecutive packet losses is 0.04%, and the probability of single packet loss is 2.44%.

To assess the computational costs of the proposed algorithm, we computed that the running time of the algorithm is 0.277 s in the reliable channel.

On the whole, the reinforcement learning algorithm can quickly converge to the Nash equilibrium policies and the probability of packet loss of two consecutive attacks is less than that of a single attack.

6.2. Case 2: Simulation Example for Unreliable Channel

Consider the same system as in Case 1, where data are transmitted over an unreliable channel, i.e., packet loss may occur due to a variety of reasons. Setting the packet loss rate as $p = \{p_1, p_2, p_3, p_4\} = \{0.01, 1, 0.001, 0.1\}$. With this setting, the packet loss rate in different states can be expressed as:

$$\xi_k = \begin{cases} 1\%, & \text{if } a_k^s = 0, a_k^a = 0 \\ 100\%, & \text{if } a_k^s = 0, a_k^a = 1 \\ 0.1\%, & \text{if } a_k^s = 1, a_k^a = 0 \\ 10\%, & \text{if } a_k^s = 1, a_k^a = 1. \end{cases}$$

The variation of $Tr(P(k))$ is plotted, as shown in Figure 5. It can be seen that the frequency of data packet loss in the unreliable channel is greater than that in the reliable channel. Furthermore, in 501–5000 iterations, the maximum number of consecutive packet losses is 2 as well.

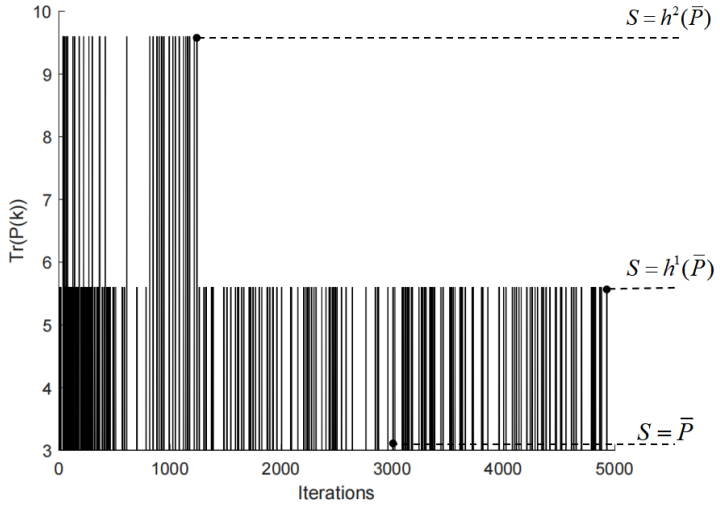

Figure 5. Variation of $Tr(P(k))$ for unreliable channel.

The learning process of $\widetilde{Q}(s, a^s, a^a)$ at state $s = \{\bar{P}, h(\bar{P}), h^2(\bar{P})\}$ is shown in Figure 6. Q-value matrix converges at about 500 steps. The convergence curve fluctuates slightly between 500 and 1300 steps which is because the existence of ϵ preserves a certain probability of random exploration. After 1300 steps, the wave disappears. According to Figure 6, it can be concluded that the elements of the Q-value matrix decrease and converge to $\widetilde{Q}^*(s, a^s, a^a)$ by the Q-learning algorithm, and the Q-value matrix obtained by convergence is shown in Table 3.

Table 3. $\widetilde{Q}^*(s, a^s, a^a)$ for unreliable channel.

State-Action	$a = (0,0)$	$a = (0,1)$	$a = (1,0)$	$a = (1,1)$
$s = \bar{P}$	7.565	2.163	13.344	5.859
$s = h(\bar{P})$	7.605	9.620	13.352	6.659
$s = h^2(\bar{P})$	7.682	12.820	13.364	7.819

There also exist Nash equilibrium points of the game based on game theory, which are shown in Table 4. The solution is conducted with the linear programming method to obtain the mixed strategy Nash equilibrium of the sensor and the attacker, denoted by π_1^* and π_2^*, respectively. It can be seen that, under channel unreliability conditions, sensors are more focused on secure transmission so as to reduce the probability of packet loss.

Table 4. Nash equilibrium π_1^* and π_2^* for unreliable channel.

State-π^*	π_1^*	π_2^*
$s = \bar{P}$	$(1,0)$	$(1,0)$
$s = h(\bar{P})$	$(0.769, 0.231)$	$(0.340, 0.660)$
$s = h^2(\bar{P})$	$(0.519, 0.481)$	$(0.468, 0.532)$

$\pi_1^* = (Pr(a^s = 0), Pr(a^s = 1))$, $\pi_2^* = (Pr(a^a = 0), Pr(a^a = 1))$.

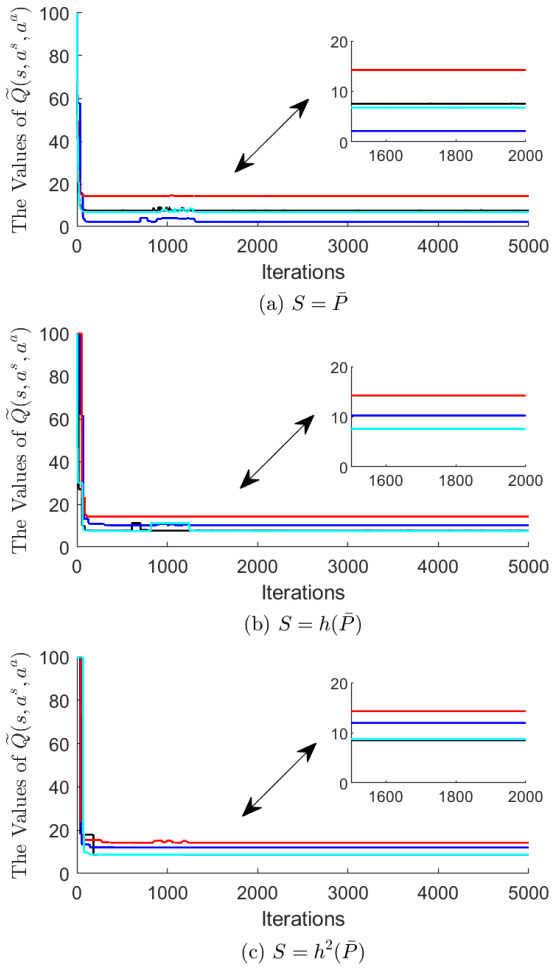

Figure 6. Learning process of $Q(s, a^s, a^a)$ for unreliable channel.

Likewise, after obtaining the optimal policy for the sensor and the attacker, the probability of packet loss under a single attack and two consecutive attacks are calculated as an evaluation criterion for the algorithm. The probability of two consecutive packet losses is 0.42%, and the probability of single packet loss is 3.76%. Compared with the results of case 1, it can be seen that the probability of packet loss in the unreliable channel is significantly higher than that in the reliable channel.

In the unreliable channel, we also evaluate the computational cost of the proposed algorithm, and the running time of the algorithm is 0.489 s.

In conclusion, the reinforcement learning algorithm can also quickly converge to the Nash equilibrium policies under the unreliable channel. The probability of losing packets in two consecutive attacks is less than that in a single attack as well and the probability of packet loss increases compared with the results of case 1.

7. Conclusions

This paper studies the security state estimation and optimal policy selection algorithm of attackers and sensors of CPSs under DoS attacks. A two-player zero-sum deterministic game is employed to describe the interactive decision process between the sensor and the attacker. A Kalman filter and reinforcement learning algorithm are introduced to evaluate policy influence and dynamically adjust strategies. Furthermore, the paper studies how to reach the Nash equilibrium of both sides to guide their decision-making choices. In consideration of the different packet loss rates under different channels in practical application, this paper proposes a reinforcement learning method based on Q-learning in two scenarios: reliable channel and unreliable channel. Simulation results of reliable and unreliable channels show that the Q-value matrix can converge in a finite number of steps

to obtain Nash equilibrium policies for both sides, which proves the effectiveness of the algorithm. The convergence process, Nash equilibrium strategy and packet loss probability under reliable and unreliable channels are elaborated and analyzed, which can guide the selection of attack and defense strategies in practical CPSs. Moreover, based on the consideration of the limited bandwidth and energy of the system, this paper optimizes the resource allocation. Specifically, the attacker and the sensor make the best response to the other's behavior by the algorithm proposed in this paper. In this way, unnecessary resource consumption caused by confrontation can be reduced, thus achieving the optimization of system resource utilization. Future work includes extending reinforcement learning algorithms under reliable and unreliable channels to multi-sensor single-attacker or multi-sensor multi-attacker security problems. In addition, the model expansion under different attacks will also be our future consideration, such as how to coordinate multiple attack sources and sensors under DDoS attacks to make optimal decisions.

Author Contributions: Writing—review and editing, Z.J.; writing— original draft, S.Z.; methodology, Y.H.; project administration, Y.Z.; supervision, C.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 62003275, Fundamental Research Funds for the Central Universities of China with Grant 31020190QD039, Ningbo Natural Science Foundation with Grant 2021J046.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are included within the article.

Acknowledgments: The authors also express great gratitude to the research team and the editors for their help.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Salau, B.; Rawal, A.; Rawat, D.B. Recent Advances in Artificial Intelligence for Wireless Internet of Things and Cyber-Physical Systems: A Comprehensive Survey. *IEEE Internet Things J.* **2022**. [CrossRef]
2. Ding, D.; Han, Q.L.; Ge, X.; Wang, J. Secure state estimation and control of cyber-physical systems: A survey. *IEEE Trans. Syst. Man, Cybern. Syst.* **2020**, *51*, 176–190.
3. Alipour-Fanid, A.; Dabaghchian, M.; Wang, N.; Jiao, L.; Zeng, K. Online-learning-based defense against jamming attacks in multichannel wireless CPS. *IEEE Internet Things J.* **2021**, *8*, 13278–13290.
4. Duo, W.; Zhou, M.; Abusorrah, A. A Survey of Cyber Attacks on Cyber Physical Systems: Recent Advances and Challenges. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 784–800.
5. Dibaji, S.M.; Pirani, M.; Flamholz, D.B.; Annaswamy, A.M.; Johansson, K.H.; Chakrabortty, A. A systems and control perspective of CPS security. *Annu. Rev. Control* **2019**, *47*, 394–411.
6. Kordestani, M.; Saif, M. Observer-based attack detection and mitigation for cyberphysical systems: A review. *IEEE Syst. Man Cybern. Mag.* **2021**, *7*, 35–60.
7. Li, T.; Chen, B.; Yu, L.; Zhang, W.A. Active security control approach against DoS attacks in cyber-physical systems. *IEEE Trans. Autom. Control* **2020**, *66*, 4303–4310.
8. Mahmoud, M.S.; Hamdan, M.M.; Baroudi, U.A. Modeling and control of cyber-physical systems subject to cyber attacks: A survey of advances and challenges. *Neurocomputing* **2019**, *338*, 101–115.
9. Alsulami, A.A.; Zein-Sabatto, S. Resilient Cyber-Security Approach For Aviation Cyber-Physical Systems Protection Against Sensor Spoofing Attacks. In Proceedings of the 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 27–30 January 2021; pp. 0565–0571.
10. Renganathan, V.; Fathian, K.; Safaoui, S.; Summers, T. Spoof resilient coordination in distributed and robust robotic networks. *IEEE Trans. Control Syst. Technol.* **2021**, *30*, 803–810.
11. Ashok, A.; Govindarasu, M.; Ajjarapu, V. Online Detection of Stealthy False Data Injection Attacks in Power System State Estimation. *IEEE Trans. Smart Grid* **2016**, *9*, 1636–1646.
12. Du, M.; Pierrou, G.; Wang, X. Targeted False Data Injection Attack against DC State Estimation without Line Parameters. In Proceedings of the 2021 IEEE Power & Energy Society General Meeting (PESGM), Washington, DC, USA, 26–29 July 2021; pp. 1–5.

13. Choraria, M.; Chattopadhyay, A.; Mitra, U.; Strom, E. Design of false data injection attack on distributed process estimation. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 670–683.
14. Li, Z.; Zhou, C.; Che, W.; Deng, C.; Jin, X. Data-Based Security Fault Tolerant Iterative Learning Control under Denial-of-Service Attacks. *Actuators* **2022**, *11*, 178.
15. Liu, W.; Sun, J.; Wang, G.; Bullo, F.; Chen, J. Resilient Control under Quantization and Denial-of-Service: Co-designing a Deadbeat Controller and Transmission Protocol. *IEEE Trans. Autom. Control.* **2021**. [CrossRef]
16. Liu, Y.; Yang, G.H. Event-Triggered Distributed State Estimation for Cyber-Physical Systems Under DoS Attacks. *IEEE Trans. Cybern.* **2022**, *52*, 3620–3631.
17. Liu, R.; Hao, F.; Yu, H. Optimal SINR-based DoS attack scheduling for remote state estimation via adaptive dynamic programming approach. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *51*, 7622–7632.
18. Hasnat, M.A.; Rahnamay-Naeini, M. A data-driven dynamic state estimation for smart grids under DoS attack using state correlations. In Proceedings of the 2019 North American Power Symposium (NAPS), Wichita, KS, USA, 13–15 October 2019; pp. 1–6.
19. Feng, S.; Cetinkaya, A.; Ishii, H.; Tesi, P.; De Persis, C. Networked control under DoS attacks: Tradeoffs between resilience and data rate. *IEEE Trans. Autom. Control* **2020**, *66*, 460–467.
20. Wang, L.; Cao, X.; Zhang, H.; Sun, C.; Zheng, W.X. Transmission scheduling for privacy-optimal encryption against eavesdropping attacks on remote state estimation. *Automatica* **2022**, *137*, 110145.
21. Yuan, H.; Xia, Y.; Yang, H. Resilient state estimation of cyber-physical system with multichannel transmission under DoS attack. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *51*, 6926–6937.
22. Pirani, M.; Nekouei, E.; Sandberg, H.; Johansson, K.H. A Graph-Theoretic Equilibrium Analysis of Attacker-Defender Game on Consensus Dynamics Under $\mathcal{H}_2$ Performance Metric. *IEEE Trans. Netw. Sci. Eng.* **2020**, *8*, 1991–2000.
23. Kurt, M.N.; Ogundijo, O.; Li, C.; Wang, X. Online cyber-attack detection in smart grid: A reinforcement learning approach. *IEEE Trans. Smart Grid* **2018**, *10*, 5174–5185.
24. Ding, K.; Ren, X.; Quevedo, D.E.; Dey, S.; Shi, L. DoS attacks on remote state estimation with asymmetric information. *IEEE Trans. Control Netw. Syst.* **2018**, *6*, 653–666.
25. Dahiya, A.; Gupta, B.B. A reputation score policy and Bayesian game theory based incentivized mechanism for DDoS attacks mitigation and cyber defense. *Future Gener. Comput. Syst.* **2021**, *117*, 193–204.
26. Li, Y.; Quevedo, D.E.; Dey, S.; Shi, L. SINR-Based DoS Attack on Remote State Estimation: A Game-theoretic Approach. *IEEE Trans. Control Netw. Syst.* **2016**, *4*, 632–642.
27. Wang, X.F.; Sun, X.M.; Ye, M.; Liu, K.Z. Robust Distributed Nash Equilibrium Seeking for Games Under Attacks and Communication Delays. *IEEE Trans. Autom. Control* **2022**. [CrossRef]
28. Xue, L.; Cao, X.; Sun, C.; Jin, S. Optimal jamming attack strategy against wireless state estimation: A game theoretic approach. In Proceedings of the IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 5989–5995.
29. Gao, W.; Deng, C.; Jiang, Y.; Jiang, Z.P. Resilient reinforcement learning and robust output regulation under denial-of-service attacks. *Automatica* **2022**, *142*, 110366.
30. Xue, L.; Sun, C.; Wunsch, D.; Zhou, Y.; Yu, F. An adaptive strategy via reinforcement learning for the prisoner's dilemma game. *IEEE/CAA J. Autom. Sin.* **2017**, *5*, 301–310.
31. He, Y.; Liang, C.; Yu, F.R.; Han, Z. Trust-Based Social Networks with Computing, Caching and Communications: A Deep Reinforcement Learning Approach. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 66–79.
32. Bozkurt, A.K.; Wang, Y.; Pajic, M. Secure planning against stealthy attacks via model-free reinforcement learning. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 10656–10662.
33. Dai, P.; Yu, W.; Wang, H.; Wen, G.; Lv, Y. Distributed reinforcement learning for cyber-physical system with multiple remote state estimation under DoS attacker. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 3212–3222.
34. Hu, J.; Wellman, M.P. Multiagent reinforcement learning: Theoretical framework and an algorithm. In Proceedings of the ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning, San Francisco, CA, USA, 24–27 July 1998; pp. 242–250.
35. Nash J.F., Jr. Equilibrium points in n-person games. *Proc. Natl. Acad. Sci. USA* **1950**, *36*, 48–49.
36. Ye, M.; Tianqing, C.; Wenhui, F. A single-task and multi-decision evolutionary game model based on multi-agent reinforcement learning. *J. Syst. Eng. Electron.* **2021**, *32*, 642–657.
37. Russell, S.J. *Artificial Intelligence a Modern Approach*; Pearson Education, Inc.: London, UK, 2010.
38. Watkins, C.J.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292.
39. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
40. Anderson, B.D.; Moore, J.B. *Optimal Filtering*; Courier Corporation: North Chelmsford, MA, USA, 2012.
41. Lyu, L.; Chen, C.; Hua, C.; Yang, B.; Guan, X. Transmission reliability enhancement for multi-sensor state estimation in industrial CPSs. In Proceedings of the 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, CA, USA, 10–14 April 2016; pp. 1057–1058.