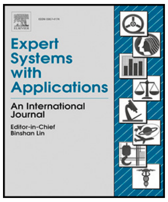
Expert Systems With Applications 210 (2022) 118394

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



GPDS: A multi-agent deep reinforcement learning game for anti-jamming secure computing in MEC network

Miaojiang Chen [a], Wei Liu [b,*], Ning Zhang [c], Junling Li [d], Yingying Ren [a], Meng Yi [e], Anfeng Liu [a,*]

[a] School of Computer Science and Engineering, Central South University, Changsha, 410083, China
[b] School of Informatics, Hunan University of Chinese Medicine, Changsha Hunan 410208, China
[c] Department of Electrical and Computer Engineering, University of Windsor, 401 Sunset, Canada
[d] Department of Electrical and Computer Engineering, University of Waterloo, 200 University Avenue, Canada
[e] School of Computer Science and Engineering, Southeast University, Nanjing, China

ARTICLE INFO

Keywords:
Deep reinforcement learning
Multi-agent
Secure computing
Decision-making
Mobile Edge Computing (MEC)

ABSTRACT

The openness of Mobile Edge Computing (MEC) networks makes them vulnerable to interference attacks by malicious jammers, which endangers the communication quality of mobile users. To achieve secure computing, the conventional method is that the mobile device reduces the attacker's malicious interference by increasing the transmission power. However, the cost of power defense is unacceptable in MEC with resource shortages. Therefore, this paper considers a novel defense strategy based on time-varying channel and describes the malicious interference countermeasure process as a multi-user intelligent game model. Because the interference model and interference strategy are unknown, this paper proposes a deep reinforcement learning multi-user random Game with Post-Decision State (named GPDS) to intelligently resist intelligent attackers. In the GPDS algorithm, mobile users need to obtain the communication quality, spectrum availability, and jammer's strategy from the state of the blocked channel. The reward of the optimal decision strategy is defined as the expected value of the maximum channel throughput, and the potential optimal channel selection strategy is obtained through Nash equilibrium. After GPDS training, mobile users can learn the optimal channel switching strategy after multi-step training. The experimental results show that the proposed GPDS achieves better anti-jamming performance, compared with SOTA algorithms.

1. Introduction

With the development of 5G network technology, mobile sensor networks provide unprecedented convenience and have a tremendous impact on the development of society (Chen et al., 2021; Huang, Leung, Liu, & Xiong, 2022; Li, Li, Liu, & Chen, 2019). However, mobile devices have the characteristics of limited resources such as small volume (Yu, Liu, Xiong, and Wang, 2022), low power (Chen, Liu, Wang, Liu and Zeng, 2021) and low computing power (Chen, Liu, Wang, Zhang et al., 2021), the network also shows considerable vulnerability. Therefore, mobile devices are also more vulnerable to malicious attacks, such as coin hopping attack (Zhu et al., 2018), data Interception (Zheng and Cai, 2020), and malicious interference (Wang et al., 2020).

For Mobile Edge Computing (MEC) networks, one serious threat is a channel interference attack. Malicious jammers reduce the communication quality of mobile users by transmitting interference signals (Mukherjee, Fakoorian, Huang, & Swindlehurst, 2014; Xiao, Liu, Li, Mandayam, & Poor, 2015). Specifically, when a mobile device is attacked by a jammer, it will keep trying to retransmit, causing the device to fall into the dilemma of rapid battery discharge and channel blocking. More seriously, interference attacks may also induce denial of service attacks (Chen & Dong, 2020; Law et al., 2009). Therefore, energy-saving and effective anti-interference technology is urgently needed for MEC networks.

To defend against interference attacks, the common technical means of wireless networks include power control (Feng and Haykin, 2019), beamforming (Tang, Wang, Zhang, Chu, and Han, 2021), relay assistance (Yang et al., 2020), and frequency hopping (Shi, An, and Li, 2021). In traditional wireless networks, power control is a powerful anti-interference strategy. Specifically, the wireless network system can use game theory to obtain the optimal anti-interference power control strategy of the devices and improve the transmission rate (Chen, Song, Xin, & Backens, 2013; Lv et al., 2017). In addition, the trusted

* Corresponding authors.
E-mail addresses: miaojiangchen@csu.edu.cn (M. Chen), weiliu@csu.edu.cn (W. Liu), ning.zhang@uwindsor.ca (N. Zhang), lijunling@cuhk.edu.cn (J. Li), yingyingRen@csu.edu.cn (Y. Ren), yimeng@seu.edu.cn (M. Yi), afengliu@mail.csu.edu.cn (A. Liu).

https://doi.org/10.1016/j.eswa.2022.118394
Received 12 April 2022; Received in revised form 26 July 2022; Accepted 2 August 2022
Available online 10 August 2022

relay is also considered to be one of the promising anti-interference technologies. Wireless networks can reduce the channel interference of jammers through beamforming and relay selection (Hoang, Duong, Suraweera, Tellambura, and Poor, 2015).

Unfortunately, many interference prevention technologies of wireless networks are not suitable for MEC networks. Both power control and beamforming and relay assistance require high hardware costs and energy consumption costs. For example, power control can greatly consume the power of mobile devices, resulting in the problem of insufficient battery life.

Among many anti-interference technologies, frequency hopping technology is applied to MEC network with its excellent anti-interference and energy-saving advantages (Huynh et al., 2018; Lee, Kim, Seo, & Har, 2019). It can enable users to switch channel spectrum and avoid potential interference. In the frequency hopping strategy, there are two kinds of users: mobile devices and jammers. When the jammer generates an interference signal, the mobile device can avoid interference by switching its transmission channel.

To defend against the unknown jamming attack model, such as the jamming strategy and the jammer transmission power are unknown, some studies use reinforcement learning technology to obtain the dynamic optimal anti-jamming strategy (Van Huynh, Nguyen, Hoang, & Dutkiewicz, 2020; Xiao et al., 2018b). Furthermore, to improve the slow convergence of traditional reinforcement learning, such as Q-learning, deep reinforcement learning technology to solve the optimal anti-interference strategy has been widely concerned.

However, many previous studies focus on the single agent anti-interference scenario, and do not fully consider the possibility of mobile device cooperation in anti-jamming. With the development of multi-agent technology, many anti-jamming studies have extended them works to multi-user scenarios (Elleuch, Pourranjbar, & Kaddoum, 2021; Yin et al., 2022). Because the previous defense technology is a passive and lagging defense means according to the configuration of attack characteristics and the issuance of defense strategies, the defense side is at a disadvantage in the anti-interference confrontation. Multi-user security Game represents the relationship between cooperation and competition among multiple users by extending the Markov game framework, so as to achieve multi-user cooperation and anti-interference to improve network security. Surprisingly, multi-agent reinforcement learning technology can realize the optimal attack and defense game in an incomplete information environment (Rowland et al., 2019; Torreño, Onaindia, & Sapena, 2015). In addition, the high-dimensional channel state and action space will slow down the training speed. Therefore, we propose a frequency hopping strategy based on multi-user anti-interference game. A deterministic gradient reinforcement learning algorithm is designed to obtain the optimal frequency hopping strategy of mobile devices in the worst case.

The main contributions are summarized as follows.

- Different from the previous discrete channel state game, we model the anti-interference countermeasure problem in MEC network as a multi-user continuous channel state game problem for the first time to reduce the high-dimensional channel state space. In addition, because the nonconvex optimization of unknown information (such as power and interference strategy) in the game model is difficult to solve in the dynamic game system, we propose a Markov decision process multi-agent model with post-state decision. The proposed algorithm does not need power control for the jamming defense to realize secure computing.
- We propose a deep reinforcement learning strategy based on the Minimax gradient strategy so that multiple mobile devices can still generalize when changing the attack strategy against jammers in the training scenario of a continuous channel game. In order to solve the problem that the minimax strategy is difficult to calculate in the objective function, we propose a user-to-user multi-user adaptive learning strategy to train samples.
- According to the experimental results, the results show that our proposed policy-based GPDS algorithm achieves better anti-jamming performance superiority compared with state-of-the-art algorithms.

The rest of this paper is organized as follows. The related work is presented in Section 2. In Section 3, we introduce the detailed anti-interference system model. The proposed GPDS algorithm is investigated in Section 4. In Section 5, the simulation results are presented and discussed. Finally, the paper is concluded in Section 6.

2. Related work

MEC network has the characteristics of broadcasting and openness (Chen, Wang, Zhang and Liu, 2021; Fan et al., 2022; Ren, Liu, Liu, Wang, & Li, 2022), which makes it extremely vulnerable to malicious interference attacks. The research on mobile device defense against malignant interference has important application significance for improving communication service quality and transmission performance.

For power control anti-interference research, D'Oro, Ekici, and Palazzo (2018) proposed a strategy of joint scheduling users and power control, and used dynamic programming to solve this kind of NP-hard problem. Xiao, Li, Dai, Dai and Poor (2018) proposed a power game strategy based on Q-learning algorithm, and derived Stackelberg Equilibrium to show the impact of multiple antennas on channel interference. Using the framework of game theory, El-Bardan, Brahma, and Varshney (2016) proposed a power control anti-interference strategy under the condition of unknown channel gain to ensure the communication service quality of legitimate users. Do, Björnson, Larsson, and Razavizadeh (2018) designed an anti-jamming machine based on power control to ensure the stability of uplink communication anti-jamming of massive multiple input multiple output systems. In the heterogeneous offloading model, Xu and Zhu (2022b) studied the problem of optimizing the monitoring mode and transmission power to maximize the average ratio of successful eavesdropping tasks under the dual constraints of transmission power and task completion time limit. Xu (2020) and Xu and Zhu (2022a) studied the problem of maximizing the successful eavesdropping probability of the monitor by jointly allocating the transmission power of the monitor and unsuspicious users.

In addition, cooperative relay beamforming is considered to be a very useful anti-jamming technology. Gu et al. (2020) modeled the joint optimization of relay selection and beamformer as MINLP, relaxed the optimization problem to a convex problem, and finally improved the anti-interference performance in the vehicle network. Aiming at the incomplete channel state information of multi-antenna eavesdroppers, Wang and Wang (2015) proposed a non convex anti-interference model combining cooperative jamming and beamforming, and solved the original problem as a convex problem. In MIMO wireless communication scenario, Liu, Li, Kong, and Zhao (2016) proposed an anti-jamming scheme based on maximum signal to jamming and noise ratio (SJNR) transmit beamforming to cancel the jamming signal of the jammer.

The above anti-interference methods require high hardware costs and energy consumption costs. Therefore, frequency hopping technology without additional hardware cost is recognized as a promising anti-interference technology in mobile networks with limited resources. Liang, Cheng, Zhang, and Zhang (2018) proposed a new mode frequency hopping technology, which combines traditional frequency hopping and mode frequency hopping technology to reduce external malicious interference and improve the security performance of communication. Gao, Xiao, Wu, Xiao, and Shao (2018) proposed a dual matrix game strategy to simulate the attack and defense process of users and jammers, and further gave the optimal anti-jamming solution under Nash equilibrium. Hanawal, Abdel-Rahman, and Krunz (2017) modeled the attack and defense of devices and jammers as a Markov game, and derived the Markov game equilibrium to obtain the optimal frequency hopping threshold to avoid malicious interference attacks.

In dynamic networks, defenders usually have unknown information such as attack strategy and interference power. To deal with such problems, reinforcement learning technology has been used to optimize dynamic wireless attack and defense countermeasures. Xiao et al. (2018c) proposed a Q-learning algorithm to improve the communication anti-interference performance of the vehicle and ensure the safe driving of the vehicle when the vehicle does not know the interference information. Additionally, Yao and Jia (2019) proposed a cooperative anti-interference intelligent algorithm based on Q-Learning to resist external malicious attacks. To solve the disadvantage of slow convergence of reinforcement learning, Van Huynh, Nguyen, Hoang, and Dutkiewicz (2019) proposed a deep reinforcement learning strategy, which enables the devices to quickly obtain the optimal channel switching strategy to improve the security performance of jammer intelligent interference.

However, many of the above researches mainly use physical layer security technology to ensure the information security in the unloading phase. Compared with these works, the multi-agent security game algorithm we consider has lower cost and more flexible operation. Note that if the continuous power and channel parameters are discretized, in the time-varying wireless channel environment, the channel state space and action space under the reinforcement learning model will become very large, and the training time will be very long to achieve the convergence effect. Therefore, we will propose a policy based deep reinforcement learning algorithm to accelerate the attack and defense learning speed under continuous variables. In addition, previous work has focused on finding the game between single users, such as the defense strategy between a single transmitter (device) and a single jammer. To enhance the anti-interference performance between mobile users, we consider a multi-user attack and defense strategy. Mobile devices can share defense information for intelligent learning to improve the security performance of communication.

3. System model

Frequency hopping is a promising technology, which has been used to release spectrum congestion. Specifically, the system has a set of frequency channels to exploit. At each time slot, each attacker can select one of the channels to send as an attacking signal, and the defender must select the appropriate channel switching action according to the current unknown attack strategy and channel state to ensure its transmission security (Jia et al., 2018).

In MEC networks, the attacker transmits interference signals to mobile users to destroy the transmission quality of legitimate users. To enhance the anti-interference ability of mobile devices, an intelligent reflecting surface (Di Renzo et al., 2020; Huang, Zappone, Alexandropoulos, Debbah, & Yuen, 2019; Wu & Zhang, 2019) auxiliary system is considered. To enhance the anti interference ability of mobile devices, an intelligent reflecting surface auxiliary system is considered in this paper. The edge network deploys an intelligent reflecting surface composed of L reflecting elements and U antennas, which can provide more powerful communication performance for mobile devices. When a malicious jammer transmits an interference signal, the intelligent reflecting surface enhances the signal power through reflection beamforming to reduce the interference.

Let $L\{1,\dots,l\}$ represent the reflecting elements set, and $N = \{1,\dots,n\}$ denote the mobile devices set. Let $h^H_{bm,n}$, $h^H_{rm,n}$, $h_{br} \in G^{L\times U}$ represent channel gain between edge base station and mobile device n, between reflecting element and mobile device n, between edge base station and reflecting element. Let $h^H_{jm,n}$ represents the channel gain between mobile device n and jammer. Let $\Psi = diag(\Psi_1,\dots,\Psi_L) \in G^{L\times U}$ represent the reflection coefficient matrix, where $\Psi_l = \eta_l e^{\Theta_l}$, $\eta_l \in [0,1]$ and $\Theta_l \in [0,2\pi]$. The important notations used throughout this article are summarized in Table 1.

Table 1
Key notations of the system model.

Notation	Description
L	Reflecting element set
h	Channel gain
Ψ	Reflection coefficient matrix
$\mathbf{y}$	Transmitted signal
X_n	Received signal for mobile device n
λ_n	Signal to interference plus noise ratio
W	Bandwidth
S	State set
A	Action set
R_i	Total reward of agent i
O	Attacker action set
r_i	Instant reward of agent i
π	Learning policy
λ_i	Learning rate
c_p	Penalty factor
c_s	Sensing cost
$Q(a,s)$	State–action value function
$L(\theta)$	Entropy loss function
$G_e(\mu_i)$	Entire objective
ζ_m	Cost factor of adaptive learning
D^k_i	Replay buffer
$P(s'\|s,a,o)$	Transition function, from s to s'
t	Time slot
$E_{\tilde{s}}[R(s,a,o)]$	Expected reward
$Q^{\pi}_i(x,a_i,\dots,a_N)$	Centralized action-value function
σ_n	Gaussian noise
N	Mobile devices set
P_n	Transmit power for mobile device n
$\mathbf{b}_n$	Beamforming vector
I_n	Interference signal at device n
$Q^*(s,a)$	Optimal state–action value function
δ	Discount factor

The transmitted signal of the base station is defined as:

$$\mathbf{y} = \sum_{n=1}^{N} \sqrt{P_n}\mathbf{b}_n z_n, \tag{1}$$

where P_n is the transmit power for mobile device n, $\mathbf{b}_n \in G^{L\times 1}$ is beamforming vector for mobile device n, $z_n \in G$ is transmitted symbol for mobile device n, $E\{|b_n|^2\} = 1$ and $E\{b_n\} = 0$. We consider the case of malicious jammers interfering with the base station by the interference signal I_n at mobile device n. Let $P_{jn} = Tr(I_n I_n^H)$ represent the power of faked signal, where $Tr(.)$ and $(.)^H$ denote the trace and conjugate transpose operations. Therefore, the total received signal for mobile device n can be written as:

$$X_n = \underbrace{\left(\mathbf{h}^H_{rm,n}\boldsymbol{\Psi}\mathbf{G} + \mathbf{h}^H_{bm,n}\right)\sqrt{P_n}\mathbf{b}_n z_n}_{\text{desired signal}} + \underbrace{\sum_{i\in N,i\neq n}\left(\mathbf{h}^H_{rm,n}\boldsymbol{\Psi}\mathbf{G} + \mathbf{h}^H_{bm,n}\right)\sqrt{P_i}\mathbf{b}_i z_i}_{\text{user interference}} + \underbrace{\sqrt{P_{jn}}\mathbf{h}^H_{jm,n}\mathbf{I}_n}_{\text{jamming signal}} + \sigma_n, \tag{2}$$

where σ_n is the Gaussian noise with variance ϑ^2_n at mobile device n. In Eq. (2), the total signal received by the device includes not only the signal expected to be received by itself, but also the interference between mobile devices and the malicious interference of jammers. Therefore, the signal to interference plus noise ratio (SINR) γ_n of mobile device n is:

$$\lambda_n = \frac{P_n\left|\left(\mathbf{h}^H_{rm,n}\boldsymbol{\Psi}\mathbf{G}+\mathbf{h}^H_{bm,n}\right)\mathbf{b}_k\right|^2}{\sum_{i\in N,i\neq n} P_i\left|\left(\mathbf{h}^H_{rm,n}\boldsymbol{\Psi}\mathbf{G}+\mathbf{h}^H_{bm,n}\right)\mathbf{b}_i\right|^2 + P_{jn}\left|\mathbf{h}^H_{jm,n}\mathbf{I}_n\right|^2 + \sigma^2_n}, \tag{3}$$

Note that when the jammer interferes with the channel, the mobile device can still transmit data, but the transmission rate will be greatly reduced. The purpose of anti-interference game is to enable mobile devices to obtain the maximum transmission rate. Accordingly, the

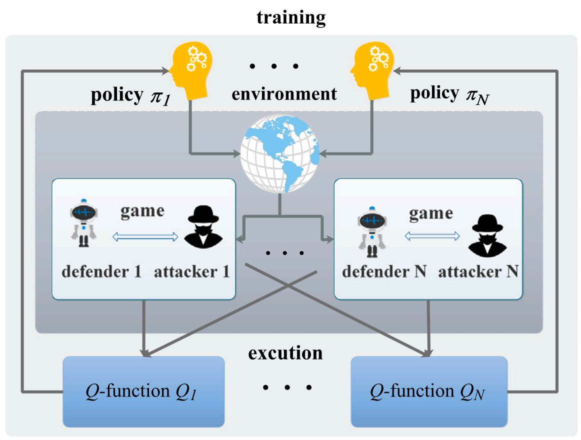


Fig. 1. Multi-agent security game in dynamic environments.

optimization problem can be defined as:

$$\begin{aligned} &\max_{\{P_n\}_{n\in N},\Psi} \textstyle\sum_{n\in N} W\log_2\left(1+\lambda_n\right)\\ &\text{ s.t. (c1)}: \textstyle\sum_{n=1}^{N} P_n \le P_{\max}\\ &\quad\text{(c2)}: \lambda_n \ge \lambda_n^{\min}, \forall n \in N\\ &\quad\text{(c3)}: \left|\Psi_m\right| = 1, 0 \le \theta_l \le 2\pi, \forall l \in L, \end{aligned} \tag{4}$$

where W is bandwidth, $\gamma_n^{\min}$ is minimum γ_n of mobile device n. Obviously, the optimization problem (see Function (4)) is non-convex. In function (4), variables reflection coefficient matrix Ψ, channel gain $h_n = \{h_{bm,n}^H, h_{rm,n}^H, h_{br}\}$ and P_n are coupled with each other, which makes it very difficult to solve the Function (4). Furthermore, the unknown channel state and interference strategy increase the uncertainty of the MEC system and the difficulty of solving the Function (4).

Therefore, we propose a policy-based multi-agent game algorithm to solve the above optimization problem. Firstly, we transform Function (4) into a Markov decision process (MDP). In a multi-agent game environment, the relation between the defender and attacker (adversary) is defined as $\langle S, A, O, T_r, R, R^O\rangle$, where S is state (such as, the Wireless channel gains). $A_1, \ldots, A_n$ represents a set of actions (such as, the channels and powers states). $O_1, \ldots, O_n$ stands for a set of attackers (the adversary from the defender's perspective). $R_i(s, a, o)$ denotes the reward function of the defender, $r_i : S \times A_i$ is rewards of agent i, and $R_i = \sum_{t=0}^{T} \gamma^t r_i^t$, while R^O represents reward of the attacker. $T_r : S \times A_1 \times \cdots \times A_n$ is the state transition function. As shown in Fig. 1, to choose actions, each agent (such as, defender and attacker) uses reinforcement learning algorithm to learn a policy $\pi_i : O_i \times A_i \to [0, 1]$. The defender and attacker aim to maximize their total expected reward $E[R]$ in the dynamic environment. At each time step, both the defender and the adversary observe the current state $s_i \in S$ and take action $a_i \in A$ and $o_i \in O$, respectively, in the light of their own learned policies π_i and π_i^o. Assume that the security game is zero-sum in our work, that is, the rewards of the defenders and attackers are opposite (that is, $R = -R^O$ and $r_i = -r^o$). Specifically, the state s_t in epoch t is defined as $s^t = \{h_n = \{h_{bm,n}^H, h_{rm,n}^H, h_{br}\}^{t-1}, P_n^{t-1}\}$. The action is defined as $a^t = \{h_n = \{h_{bm,n}^H, h_{rm,n}^H, h_{br}\}^t, P_n^t, \Theta_l^t\}$.

For anti-jamming attack and defense games, the most important element is the design of the reward function. For convenience, the intelligent reflecting surface is represented as IRS. The states $[on, off]$ of IRS are Markovian, $\Phi_{0,1}$ ($\Phi_{1,0}$) is the probability of IRS transiting from the state inactive (active) to state active (inactive).

At each time slot t, $T_t^S \in \{1, 0\}$ denotes the sensing token, which indicates whether the mobile user needs to transmit data on the current channel. T_r denotes the transition probability, which is the probability that the user switches to the next secure channel. $T_t^S = 0$ represents the mobile user keep current channel state and $a_t = 0$. $T_t^S = 1$ is an active sensing token, the user selects a target channel h_i and $a_t = i$. Meanwhile, $o_t = 0$ means that the channel jammer stops interference in order to avoid anti-interference by the IRS (i.e., attack penalty), or selects a target channel i' to send the faked signal ($o_t = i'$). The mobile user selects the channel with the best transmission performance (i.e. not interfered) for data transmission through the multi-user game strategy. If the mobile user is attacked in the transmission process, the IRS system of the base station will be notified to implement the anti-interference strategy to reduce the impact of the jammer attack. If the communication quality is still seriously affected after anti-interference, channel switching is carried out to avoid an interference attack.

The reward of mobile user is defined as:

$$\begin{aligned} r_n = &\underbrace{c_p T_{\{o_n>0\}} \cdot \Phi_{1,T_{t-1}^{on}}}_{\text{jamming attack penalty}} + \underbrace{\Phi_{0,T_{t-1}^{on}} \cdot W\log_2\left(1+\lambda_n\right)}_{\text{goodput}}\\ &\qquad\qquad - \underbrace{c_s T_t^S \cdot T_{\{a_n>0\}}}_{\text{sensing cost}}, \end{aligned} \tag{5}$$

where c_p is the penalty factor of the jammer's anti-jamming signal transmitted by the IRS, c_s is the sensing cost of IRS. T_{t-1}^{on} is the IRS state in $t-1$.

4. Proposed GPDS learning algorithm

In this section, we propose a novel policy based multi-user reinforcement learning game algorithm to solve the MDP optimization problem mentioned in the last section. The architecture for security game with deep reinforcement learning is shown in Fig. 2.

In our work, we derive a policy-based deep reinforcement learning game with Post Decision State (GPDS) algorithm that performs well in a multi-agent environment. Different from the existing reinforcement learning games (Xiong et al., 2020, 2019; Xiong, Zhao, Niyato, Deng and Zhang, 2020), we propose PDS to deal with the uncertainty in the process of random games to find the optimal strategy. Different decisions are made for different current decision states by assigning values to the current state decision and the resulting post decision state. Then the expectation of the sum of all current decision state values from post decision state to termination state is the PDS value. However, some constraints are as follows.

1. We do not suppose any structure in the communication approach between agents.
2. We do not suppose a differentiable environmental dynamics model.
3. The trained policies can only own their observations at performance time.

For n agents in anti-interference security game, their action sets are denoted as $A_1, \ldots, A_n$ and reward sets are denoted as $R_1, \ldots, R_n$. At time slot t, n agents perform policies $\pi_1, \ldots, \pi_n$ simultaneously, the expected reward of agent i is $R_i(\pi_1, \ldots, \pi_n)$.

The above game process can be called One Stage Policy (OSP) general sum security game, which has a Nash equilibrium. For $\pi_1, \ldots, \pi_n$, no agent can increase the expected reward by unilaterally changing its policy, that is:

$$R_i(\pi_1, \ldots, \pi_n) \ge R_i(\pi_1, \ldots, \pi_{i-1}, \pi_i', \pi_{i+1}, \ldots, \pi_n), \tag{6}$$

for all OSP policy π_i, and $1 \le i \le n$. One OSP security game can have multiple Nash equilibrium, and each Nash equilibrium has a corresponding expected reward that does not necessarily equal to the rewards of other Nash equilibrium.

Multi-agent Nash-Q function can be undated by:

$$\begin{aligned} &\tilde{Q}_{i+1}^{*}[s, a_1, \ldots, a_p, o_1, \ldots, o_m] = \\ &\begin{cases} (1-\lambda_i)Q_i^{*}(\tilde{s}, a_1, \ldots, a_p, o_1, \ldots, o_m) + \\ \lambda_i[r_i + \delta Nash_s(s, Q_1, \ldots, Q_n)], \\ for \quad (\tilde{s}, a, o) = (\tilde{s}, a_1, \ldots, a_p, o_1, \ldots, o_m), \\ Q_i^{*}(s, a, o), otherwise, \end{cases} \end{aligned} \tag{7}$$

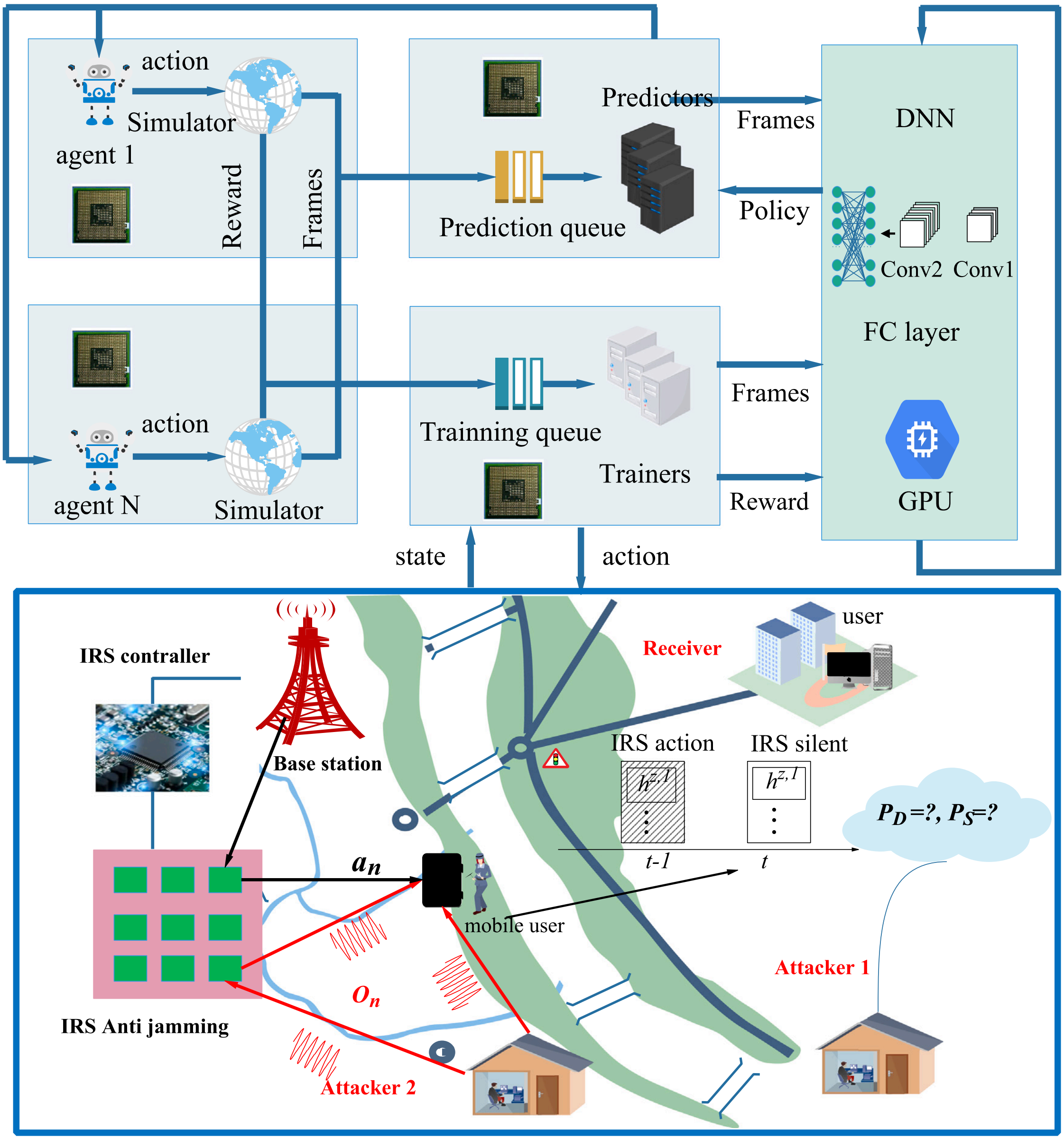


Fig. 2. The security game architecture for MEC networks.

where $m + p = n$ and $Nash_s(s, Q_1, \dots, Q_n) = Q_i(s, \pi_1, \dots, \pi_n)$ when $\pi_1, \dots, \pi_n$ are Nash equilibrium for $Q_1, \dots, Q_n$ at state s.

Meanwhile, the optimal Q function can be further defined as:

$$
\begin{aligned}
Q^*(s, a_1, \dots, a_p, o_1, \dots, o_m) = r^k(s, a_1, \dots, a_p, o_1, \dots, o_m) \\
+ \sum_{\tilde{s}} P^k(\tilde{s}|s, a_1, \dots, a_p, o_1, \dots, o_m)\tilde{Q}^*.
\end{aligned}
\tag{8}
$$

Lemma 1 (Hu, Wellman, et al., 1998). *$\pi_1, \dots, \pi_n$ and $\upsilon_1, \dots, \upsilon_n$ are two adversarial equilibria for OSP security game. If exists an adversarial equilibrium, all of its adversarial equilibria have the same value, and*

$$
\begin{aligned}
R_i(\pi_1, \dots, \pi_n) &\geq R_i(\pi_1, \dots, \pi_{i-1}, \pi_i', \pi_{i+1}, \dots, \pi_n) \\
&= R_i(\upsilon_1, \dots, \upsilon_{i-1}, \upsilon_i', \upsilon_{i+1}, \dots, \upsilon_n) \\
&\geq R_i(\upsilon_1, \dots, \upsilon_n).
\end{aligned}
\tag{9}
$$

Lemma 2 (Hu et al., 1998). *If an OSP security game has a coordination equilibrium, all of its coordination equilibria have the same value.*

For convenience, it is assumed that each agent can know what type of agent it encounters: "foe" (adversarial equilibrium) or "friend" (coordination equilibrium). For an agent i, all other agents are divided into two groups. One group is agent i's friend, which helps agent i maximize its reward return together. The other group is agent i foe, which opposes agent i and reduces agent i's reward return. Therefore, there are two groups for each agent. Thus, the general sum game of multi-agent is transformed into a zero-sum game of two agents. The solution of the Nash equilibrium strategy is given by:

$$
\begin{aligned}
&Nash_s(s, Q_1, \dots, Q_n) = \\
&\max_{\pi_1(s), \dots, \pi_n(s)} \min_{o_1, \dots, 0_p \in O_1 \times, \dots, \times O_m} \sum_{a_1, \dots, a_p \in A_1 \times, \dots, \times A_p} \\
&Q_i(s, a_1, \dots, a_p, o_1, \dots, o_m)\pi_1(s, a_1), \dots, \pi_p(s, a_p).
\end{aligned}
\tag{10}
$$

Therefore, the approximate policy is learned by action $j's$ actions, with entropy being regularized:

$$
L(\theta_i^j) = -E_{a_j, o_j}[\log \tilde{\mu}_i^j(a_j|o_j) + \lambda X(\tilde{\mu}_i^j)],
\tag{11}
$$

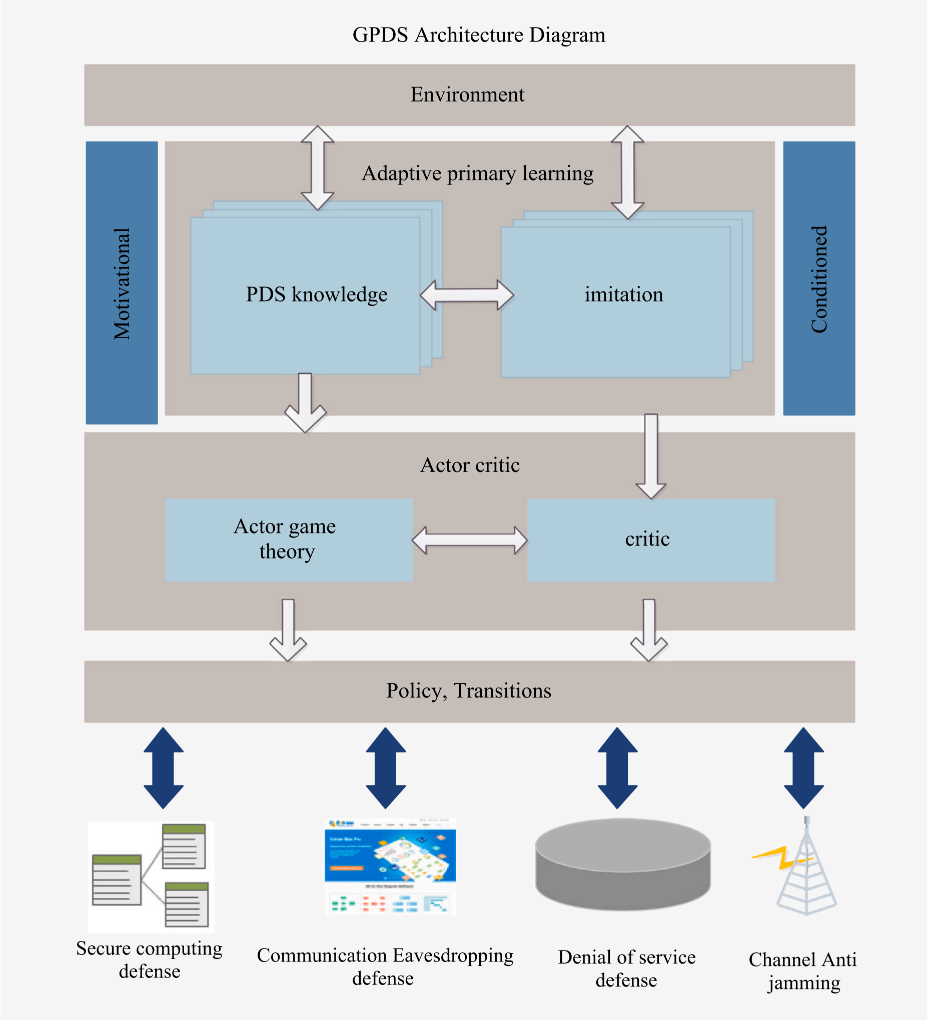


Fig. 3. The GPDS reinforcement learning architecture diagram.

where, X represents the policy distribution entropy. Furthermore, we use an approximate value $\tilde{y}$ to instead of y in formula (11) as follows:

$$\tilde{y} = r_i + \gamma Q_i^{\mu'}(x', \tilde{\mu}_1'(o_1), \ldots, \tilde{\mu}_N'(o_N)). \tag{12}$$

A recurring iteration issue in MARL is that the environment is non-stationary because of the policies changing. In particular, in an attack–defense environment, an attacker can obtain a strong policy through over-fitting to the action of its opponent. To achieve game policies, there are more changes in the strategies the defender implements in the game. In GPDS learning, we train a set of K different sub-policies, the algorithm randomly chooses different sub-policies with sub-policy k representing $\mu_{\theta_i^k}$ at each episode. For attacker i, we maximize the entire objective:

$$G_e(\mu_i) = E_{k\sim uniform(1,K), S\sim p^{\mu}, a\sim\mu_i^k}[R_i(s, a, o)]. \tag{13}$$

Because of different sub-policies performed in different episodes, the GPDS learning maintains a replay buffer D_i^k for μ_i^k of agent i. Consequently, the gradient of the entire objective w.r.t. θ_i^k is written as:

$$\nabla_{\theta_i^k} G_e(\mu_i) = \frac{1}{K} E_{x,a\sim D_i^k}[\nabla_{\theta_i^k}\mu_i^k(a_i|o_i)\nabla_{a_i}Q^{\mu_i}(x, a_1, \ldots, a_N)]. \tag{14}$$

The deep reinforcement learning architecture for security games is illustrated in Fig. 3, which contains two parts: the actor–critic component and the adaptive module Learning component. In our paper, the adaptive module Learning component is used to improve learning speed. The core idea of the adaptive module Learning is trying to combine PDS knowledge with imitation process to select the policy. The adaptive module Learning component of reinforcement learning architecture for security game is shown in Fig. 4(a), which contains two processes. In the first process, according to the Markov hypothesis and the uniform distribution of the initial solution of the security game, the agent can use PDS knowledge. In reinforcement learning, uniform distribution of policies helps agents balance exploration and development. At the beginning of the training process, a random policy selector is used to generate the exploratory random behavior of the agent from d_i. Two learning rules $L^1_{i,j|k}(t)$, $L^2_{i,j|k}(t)$ are used to estimate the result. The learning rule $L^1_{i,j|k}(t)$ is defined as:

$$L^1_{i,j|k}(t) = \frac{\eta(i,j|k)(t-1)}{\eta(i|k)(t-1)}, \tag{15}$$

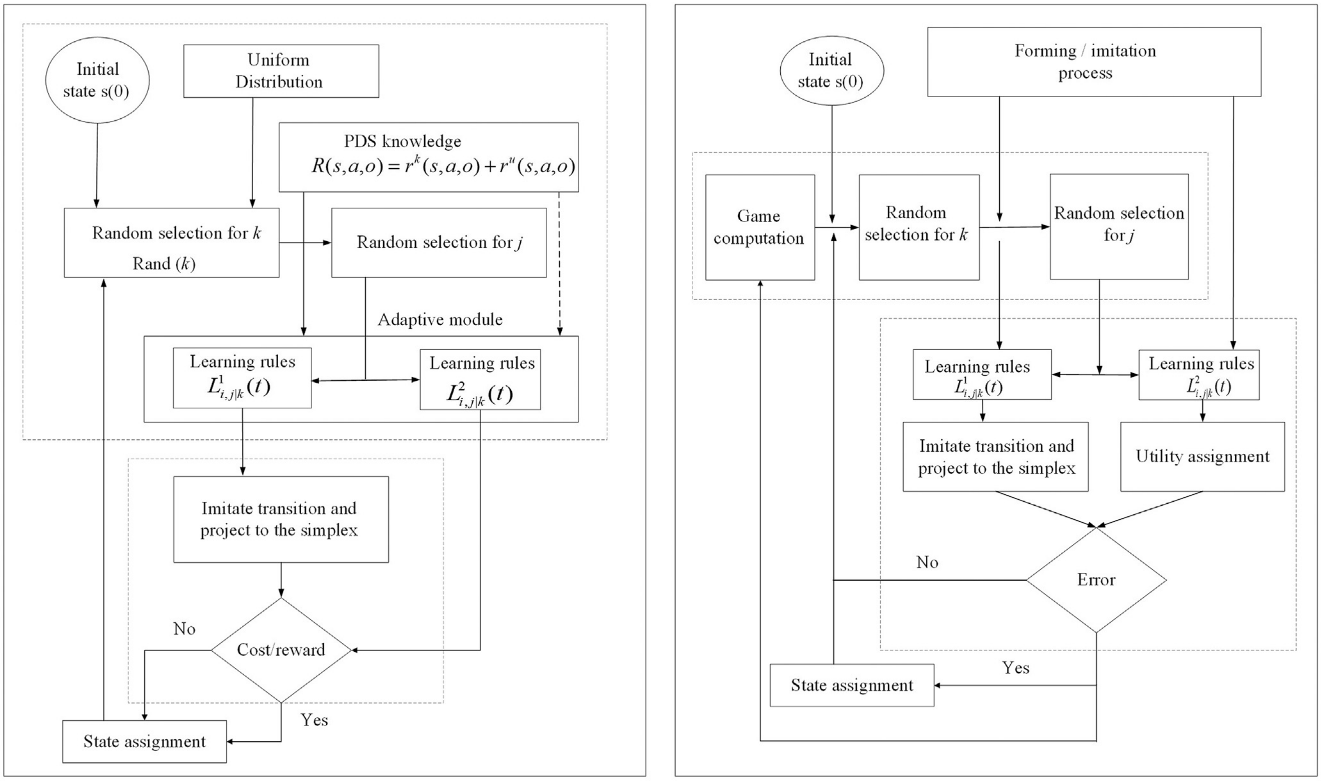
where η is defined as:

$$\eta(i|k)(t) = \sum_{m=1}^{t} F(s_m = s_i, a_n = a_k),$$

$$\eta(i,j|k)(t) = \sum_{m=1}^{t} F(s_{m+1} = s_j | s_m = s_i, a_n = a_k),$$

and

$$F(\kappa_t) = \begin{cases} 1, \text{if the even } \kappa \text{ happens at time t} \\ 0, \text{otherwise.} \end{cases}$$



(a) Adaptive primary learning architecture

(b) Actor critic architecture

Fig. 4. Adaptive primary Learning and actor–critic architecture.

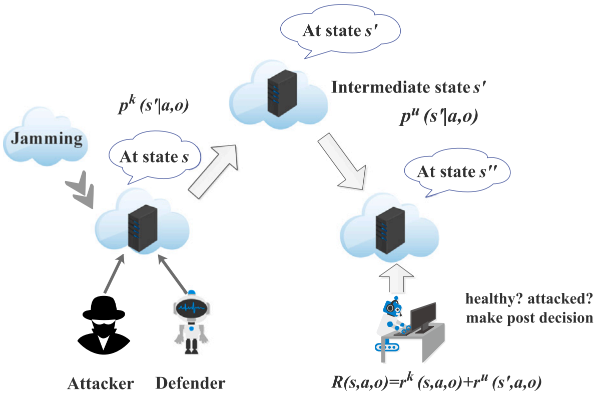


Fig. 5. The process of PDS.

The learning rule $L^2_{i,j|k}(t)$ is defined as:

$$L^2_{i,j|k}(t) = \frac{\Theta(i,j|k)(t)}{\eta(i,j|k)(t)}, \tag{16}$$

where

$$\Theta(i,j|k)(t) = \sum_{m=1}^{t} \zeta_m F\left(s_{m+1} = s_j | s_m = s_i, a_n = a_k\right),$$

and ζ_m is cost factor.

The learning process of the adaptive module is as follows. The agent i begins with s_0. As a solution of the security game, the initial state is a fixed uniform distribution of policy. After that, the agent selects randomly an action a. Then, the transition matrix is used to select the next state s_{t+1}. Finally, the learning processes started again until the algorithm converges.

Actor–critic algorithm is a temporal difference learning model. It is used to generate the policy structure. Choosing the action, the next state is called the actor, while the operation responsible for estimating the value function is called the critic. The role of the critic is to control the whole learning process and analyze whether the policy makes the agent make the best action. In order to complete the task, an error estimator is used by the critic to manage all learning decisions. The actor–critic component is shown in Fig. 4(b). In Actor–critic algorithm, we need to fit a value function Q with post decision state to get a better gradient estimation.

In this way, the policies can adopt extra information to ease the process of learning, provided this information is not adopted at test time. However, this has no application to Q-learning. Thus, a policy-based multi-agent actor–critic combined with a Post Decision State, termed GPDS is proposed. In the proposed GPDS algorithm, we suppose that given an action o of adversary at state s, the defender will take action a, the Post Decision State agent (that is, the agent who has extra information x in this algorithm) can achieve a known reward $r^k(s_i, a_i, o_i)$ and current state transits to an intermediate state $\tilde{s}$, with a known probability $p^k(s', a, o)$. However, the extra information can be unknown to the adversary. In next step, using an unknown probability $p^u(s'|s, a, 0)$, the intermediate state will transits to the next state s', meanwhile, receive a reward $r^u(s'|s, a, o)$, which depends on the Post Decision State. In particular, we suppose that the next state s' is independent. The process of the Post Decision State is shown as Fig. 5. Then we can write the transition function, that is, from s to s' as:

$$P(s'|s,a,o) = \sum_{\tilde{s}} P^u(s'|\tilde{s},a,o)P^k(\tilde{s}|s,a,0). \tag{17}$$

The expected reward of agent can be written as:

$$E_{\tilde{s}}[R(s,a,o)] = r^k(s,a,o) + \sum_{\tilde{s}} P^k(\tilde{s}|s,a,o)r^u(\tilde{s}|s,a,o). \tag{18}$$

The basic idea of the presented GPDS algorithm is that, in place of updating a single gradient of the return for agent i at a time, the agent i with the extra information on $P^k(s'|s,a,o)$ and $r^k(s,a,o)$ first learns a gradient $\nabla_{\theta_i} E[R_i]$, which $\theta = \theta_1, \ldots, \theta_N$ is policies parameterized, considering a game with N agents. $\pi = \pi_1, \ldots, \pi_N$ represent the set of all agent policies.

In this paper, combining with traditional Actor–Critic, we use a Post Decision State method to get the optimal policy. The optimal Post Decision State quality function $\tilde{Q}^*$ can be written as:

$$\tilde{Q}^* = r^{\mu}(\tilde{s},a,o) + \delta \sum_{s'} P^{u}(s'|\tilde{s},a,o)V^*(S'), \tag{19}$$

where, δ is the discount factor, V^* is the value function and Q^* is the optimal quality function in GPDS algorithm. Therefore, we have the following:

$$Q^*(s,a,o) = E_R[R(s,a,o) + \delta V^*(s')]. \tag{20}$$

$$V^*(s) = maxmin \sum_{a} Q^*(s,a,o)\pi(s,a,o). \tag{21}$$

Applying the extra information $r^k(s,a,o)$ and $p^k(\tilde{s}|s,a,o)$, the optimal quality $\tilde{Q}^*$ is further written as:

$$Q^*(s,a,o) = r^k(s,a,o) + \sum_{\tilde{s}} P^k(\tilde{s}|s,a,o)\tilde{Q}^*. \tag{22}$$

In our work, after observing the sample $(s_i, a_i, o_i, r^k(s_i,a_i,o_i), \tilde{s}_i, r^u(s_i,\tilde{a}_i,o_i), s_{i+1})$, the Post Decision State value function $\tilde{Q}^*$ is updated by:

$$\begin{aligned}\tilde{Q}^*_{i+1}(\tilde{s},a_i,o_i) &= (1-\lambda_i)\tilde{Q}^*_i(\tilde{s},a_i,o_i)\\ &+\lambda_i[r^u(\tilde{s},a_i,o_i) + \delta V^*(s_{i+1})].\end{aligned} \tag{23}$$

where $\lambda_i \in (0,1)$ is a learning rate.

Thus, the gradient function of the expected reward can be written as:

$$\nabla_{\theta_i} E[R_i] = E_{s\sim p^u, a_i\sim\pi_i}[\nabla_{\theta_i} \log \pi_i(a_i|o_i) Q_i^{\pi}(x, a_i, \ldots, a_N)]. \tag{24}$$

where $Q_i^{\pi}(x, a_i, \ldots, a_N)$ is a centralized action-value function. Furthermore, extra information x consists of the observation of all agents, $x = (o_1, \ldots, o_N)$.

We consider N continuous policies μ_{θ_i} with respect to parameters θ_i, the gradient can be defined as:

$$\begin{aligned}&\nabla_{\theta_i} G(\mu_{\theta_i}) = \nabla_{\theta_i} E[R_i]\\ &= E_{x,a_i\sim D}[\nabla_{\theta_i}\mu_i(a_i|o_i)Q_i^{\mu}(x,a_1,\ldots,a_N)|a_i = \mu_i(o_i)],\end{aligned} \tag{25}$$

where D is the memory pool buffer, which contains the data samples, saving the all the training experiences of system, x denotes the state information. Furthermore, the centralized action-value function Q_i^{μ} is written as:

$$\begin{aligned}&\omega(\theta_i) = E_{x,a,r,x'}[(Q_i^{\mu}(x,a_1,\ldots,a_N) - y)^2],\\ &\quad y = r_i + \gamma Q_i^{\mu'}(x', a_1', \ldots, a_N'),\end{aligned} \tag{26}$$

where $\mu' = \{\mu_{\theta_1'}, \ldots, \mu_{\theta_N'}\}$ is the set of objective policies, θ_i' represents the delayed parameters.

To realize the multi-agent game, we perform the Minimax operation on the training objective function:

$$\begin{aligned}G\left(\theta_i\right) &= E_{s\sim p^{\mu}, a_i\sim\pi_i}\left[R_i\right]\\ &= \min\nolimits_{a^k_{j\neq i}} E_{s\sim p^{\mu}}\left[\textstyle\sum_{t=0}^{T}\gamma^k r_i\left(s^k, a_1^k, \ldots, a_N^k\right)\Big|_{a_i^k=\mu\left(o_i^k\right)}\right]\\ &= E_{s^0\sim\rho}\left[\min\nolimits_{a^0_{j\neq i}} Q_i^{\mu}\left(s^0, a_1^0, \ldots, a_N^0\right)\Big|_{a_i^0=\boldsymbol{\mu}\left(o_i^0\right)}\right].\end{aligned} \tag{27}$$

Algorithm 1 : GPDS learning algorithm

```
Initialization: $\gamma = 0.999,\ D = 10^6,\ \epsilon = 0.995,\ N$
Select initial state $s$
while $t$ <Length(episodes) do
    The agent $i$ chooses appropriate action
    $a_i = \mu_{\theta_i}(o_i)$
    and searching perform action
    $a = (a_i, ..., a_N)$
    Observation: The next state $s_i'$ and reward $r_i$
    save $(s_i, a_i, r_i, s_i')$ in $D$
    $s' \rightarrow s$
    while $i < N$ do
        Random sampling a batch of K samples $(s_k, a_k, r_k, s_k')$ from replay buffer
        set $y^j = r_i^k + \gamma Q^{\boldsymbol{\mu}'}_{\mathrm{M},i}\left(\mathbf{s}'^k, a_1', \ldots, a_N'\right)\Big|$
        $a_i' = \boldsymbol{\mu}_i'\left(o_i^k\right), a'_{j\neq i} = \boldsymbol{\mu}_j'\left(o_j^k\right) + \hat{\chi}_j'$ with $\hat{\chi}_j$ in Eq. (31)
        $s_k' = \mu_k'(o_k')$
        minimizing the loss to update critic:
        $Loss(\theta_i) = \frac{1}{K}\sum_k (y^j - Q_i^{\mu'}(s_k', a_k', ..., a_N'))^2.$
        Applying the sampled policy gradient to update actor:
        $\nabla_{\theta_i} G \approx \frac{1}{K}\sum_k \nabla_{\theta_i}\mu_i(o_i^k)\nabla_{a_i} Q_i^{\mu}(s_k', a_k', ..., a_N').$
        $a_i = \mu_i\left(o_i^k\right), a^*_{j\neq i} = a_j^k + \hat{\chi}_j$
    end while
    For each agent $i$, using follow formula to update parameters:
    $\theta_i' \leftarrow \rho\theta_i + (1-\rho)\theta_i'.$
end while
```

Note that s^{k+1} at epoch $k+1$ depends on p^{μ}, action $\mu(o_i^k)$, and previous adversarial actions $a^k_{j\neq i}$. According to Eq. (27), modified Q function can be defined as:

$$\begin{aligned}&Q_i^{\boldsymbol{\mu}}\left(s, a_1, \ldots, a_N\right) = r_i\left(s, a_1, \ldots, a_N\right) +\\ &\gamma E_{s^k}\left[\min\nolimits_{a^k_{j\neq i}} Q_i^{\boldsymbol{\mu}}\left(s^k, a_1^k, \ldots, a_N^k\right)\Big|_{a_i^k=\boldsymbol{\mu}_i(s^k)}\right].\end{aligned}$$

Importantly, for each user (agent) i, all adversarial actions do not depend on its θ_i. Using this property, we calculate $\nabla_{\theta_i} G(\mu_{\theta_i})$ by the deterministic gradient rule. Therefore, we further modify Eq. (25) as follows:

$$\nabla_{\theta_i} G\left(\theta_i\right) = E_{\mathbf{x}\sim D}\begin{bmatrix}\nabla_{a_i} Q_i^{\mu}\left(\mathbf{x}, a_1^*, \ldots, a_i, \ldots, a_N^*\right)\nabla_{\theta_i}\boldsymbol{\mu}_i\left(o_i\right)\\ a_i = \boldsymbol{\mu}_i\left(o_i\right)\\ a^*_{j\neq i} = \arg\min\nolimits_{a_{j\neq i}} Q_i^{\mu}\left(\mathbf{x}, a_1, \ldots, a_N\right)\end{bmatrix}. \tag{28}$$

Therefore, Eq. (26) introduces a minimum update Q rule:

$$\begin{aligned}\omega\left(\theta_i\right) &= E_{\mathbf{x},a,r,\mathbf{x}'\sim D}\left[\left(Q_i^{\boldsymbol{\mu}}\left(\mathbf{x}, a_1, \ldots, a_N\right) - y\right)^2\right]\\ y &= r_i + \gamma Q_i^{\boldsymbol{\mu}'}\left(\mathbf{x}', a'^{\star}1, \ldots, a_i', \ldots, a'^{\star}N\right)\\ a_i' &= \boldsymbol{\mu}_i'\left(o_i\right)\\ a'^{*}_{j\neq i} &= \arg\min_{a'_{j\neq i}} Q_i^{\boldsymbol{\mu}'}\left(\mathbf{x}', a_1', \ldots, a_N'\right).\end{aligned} \tag{29}$$

Note that a difficult problem in the interference game in this paper is how to deal with the minimization operation in Eqs. (28) and (29). Because the channel state space is nonlinear, the computational cost of gradient descent using the inner loop is too high. Therefore, we decided to use user-to-user multi-user adversarial learning for training. The core ideas are as follows: (1) The local linear function is used to approximate the nonlinear function in the original problem. (2) Multi-step gradient descent is used to replace the internal loop minimization in the original problem.

Table 2
Neural network architecture.

Network	Units	Memory	Activation	Layer
Critic	100	2048	ReLU	Fully connected
Critic	100	2048	Tanh	Fully connected
Critic	100	2048	ReLU	Fully connected
Critic	100	2048	ReLU	Fully connected
Critic	1	1024	Sigmoid	Fully connected
Actor	100		ReLU	Fully connected
Actor	80		ReLU	Fully connected
Actor	20		ReLU	Fully connected
Actor	1	2048	Sigmoid	Fully connected

Table 3
The parameters of GPDS algorithm.

Parameters	Value
Learning rate	0.0005
Explore start	0.05
Explore stop	0.01
Explore decay rate	0.0001
Discount factor	0.95
Memory size	3000
Episodes	10 000
Batch size	128
The bandwidth	5 MHz
Noise factor	0.1

Based on the above analysis, Eq. (29) is redefined as follows with auxiliary variables χ:

$$\begin{aligned}
\omega\left(\theta_i\right) &= E_{\mathbf{x},a,r,\mathbf{x}'\sim D}\left[\left(Q_i^{\boldsymbol{\mu}}\left(\mathbf{x},a_1,\ldots,a_N\right)-y\right)^2\right]\\
y &= r_i+\gamma Q_i^{\boldsymbol{\mu}'}\left(\mathbf{x}',a_1'^{*},\ldots,a_i',\ldots,a_N'^{*}\right)\\
a_j'^{*} &= a_j'+\chi_j,\quad \forall j\neq i\\
a_t' &= \boldsymbol{\mu}_t'\left(o_t\right),\quad \forall 1\leq t\leq N\\
\chi_{j\neq i} &= \arg\min_{\chi_{j\neq i}} Q_i^{\boldsymbol{\mu}'}\left(\mathbf{x}',a_1'+\chi_1,\ldots,a_i',\ldots,a_N'+\chi_N\right).
\end{aligned}\tag{30}$$

The core of Eq. (30) is to find a group of interference χ, so that the interference action a'^{*} can minimize the Q value. We first linearize the Q function at $Q_i^{\boldsymbol{\mu}}\left(\mathbf{x},a_1,\ldots,a_N\right)$, and then we can obtain the local approximation of the desired χ by gradient descent:

$$\hat{\chi}_j=-\sigma\nabla_{a_j}Q_i^{\boldsymbol{\mu}'}\left(\mathbf{x}',a_1',\ldots,a_j,\ldots,a_N'\right),\forall j\neq i,\tag{31}$$

where σ is tunable coefficient, that is, the step size of gradient descent.

Combining Eqs. (28) and (31), we have:

$$\nabla_{\theta_i}G\left(\theta_i\right)=
E_{\mathbf{x},a\sim\mathcal{D}}\begin{bmatrix}\nabla_{\theta_i}\boldsymbol{\mu}_i\left(o_i\right)\nabla_{a_i}Q_i^{\mu}\left(\mathbf{x},a_1^{*},\ldots,a_i,\ldots,a_N^{*}\right)\mid\\ a_i=\boldsymbol{\mu}_i\left(o_i\right)\\ a_j^{*}=a_j+\hat{\chi}_j,\quad\forall j\neq i\\ \hat{\chi}_j=-\sigma_j\nabla_{a_j}Q_i^{\mu}\left(\mathbf{x},a_1,\ldots,a_N\right)\end{bmatrix},$$

and

$$\begin{aligned}
\omega\left(\theta_i\right) &= E_{\mathbf{x},a,r,\mathbf{x}'}\left[\left(Q_i^{\boldsymbol{\mu}}\left(\mathbf{x},a_1,\ldots,a_N\right)-y\right)^2\right]\\
y &= r_i+\gamma Q_i^{\boldsymbol{\mu}'}\left(\mathbf{x}',a'^{*}1,\ldots,a_i',\ldots,a'^{*}N\right)\\
a_k' &= \boldsymbol{\mu}_k'\left(o_k\right),\quad\forall 1\leq k\leq N\\
a_j'^{\star} &= a_j'+\hat{\chi}_j',\quad\forall j\neq i\\
\hat{\chi}_j' &= -\sigma_j\nabla_{a_j'}Q_i^{\boldsymbol{\mu}'}\left(\mathbf{x},a_1',\ldots,a_N'\right).
\end{aligned}\tag{32}$$

The detailed execution process of the GPDS algorithm is shown in Algorithm 1.

Remark. When a game has multiple equilibria (Nash or other equilibrium concepts), it is quite common and reasonable that in many applications (e.g., economics, operations research), we assume that there is a given and fixed equilibrium selection rule or some tie-breaking rule such that we do not have to worry about adversarial equilibrium selection. In other words, we assume that there is some kind of agreement between (self-interested, non-cooperative) players that they would choose a certain single equilibrium.

5. Simulation results and analysis

5.1. Environment settings

In this section, we implement the GPDS algorithm to justify its effectiveness. The Tensorflow 2.1.0 and CUDA Toolkit (NVIDIA deep learning library) are employed to implement our proposed GPDS algorithm on CentOS 7. The hardware environment is Intel(R) Gold 624 @ 2.60 GHz CPU and NVIDIA Tesla V100 32G GPU × 4. The deep neural network architecture and parameters are shown in Tables 2, 3, respectively.

We compare the proposed GPDS approach with the following algorithms:

- Deep Q-network (DQN) (Gao, Qin, Jing, Ni, and Jin, 2019): The single agent deep enhancement algorithm can only deal with discrete state and action space.
- Actor–Critic (Lee, Nagabandi, Abbeel, and Levine, 2020): Actor–Critic algorithm is a combination of policy-based and value-based methods. The actor net is responsible for generating actions and interacting with the environment. Critic net is responsible for evaluating the actor's performance and guiding the actor's actions in the next stage.
- Deep Deterministic Policy Gradient algorithm (DDPG) (Yu et al., 2021): This is a single agent deep reinforcement learning algorithm based on actor–critic architecture. An actor is used to make up for the disadvantage that DQN algorithm cannot deal with continuous control problems.

In the communication model, the channel state value is a continuous value that changes with time. However, DQN algorithm is only suitable for discrete state, Therefore, we discretize the attack and defense model in DQN algorithm. For DQN, we considers 200 channels, each continuous channel state value is divided into more than 30 000 channel gain discrete points.

5.2. Result analysis

We show the loss curves over 10 000 episodes for four algorithms in Fig. 6. It can be seen that the curve does not decline smoothly, because the input data in RL changes step by step, and different data will be obtained according to the learning situation. Observing the loss curves, the GPDS algorithm converges downward. About 2000 episodes, although the loss curves of GPDS do not show a downward convergence, but continuous training can converge. However, the stability of traditional DRL methods is less than GPDS.

The reward performance of the proposed GPDS is compared in Fig. 7. We show the learning curves over 10 000 episodes for four algorithms. Traditional algorithms, such as DDPG, Actor–Critic and DQN, do not perform as well as GPDS.

We evaluate the effectiveness of learning the policies of other agents, and the result is shown in Fig. 8. In particular, learning the policies of other agents perfectly and learning with approximated policies can achieve the same average reward as adopting the true policy, and as we can see, without an obvious slowdown in convergence.

Additionally, we measured the performance by the following formula:

$$\Phi(n)=[\tilde{r}_{PDS}(n)-\tilde{r}(n)]/average(\tilde{r}(n)).\tag{33}$$

The relative performance comparison between this paper algorithm, DDPG, Actor–Critic, and DQN algorithms is shown in Fig. 9. As we

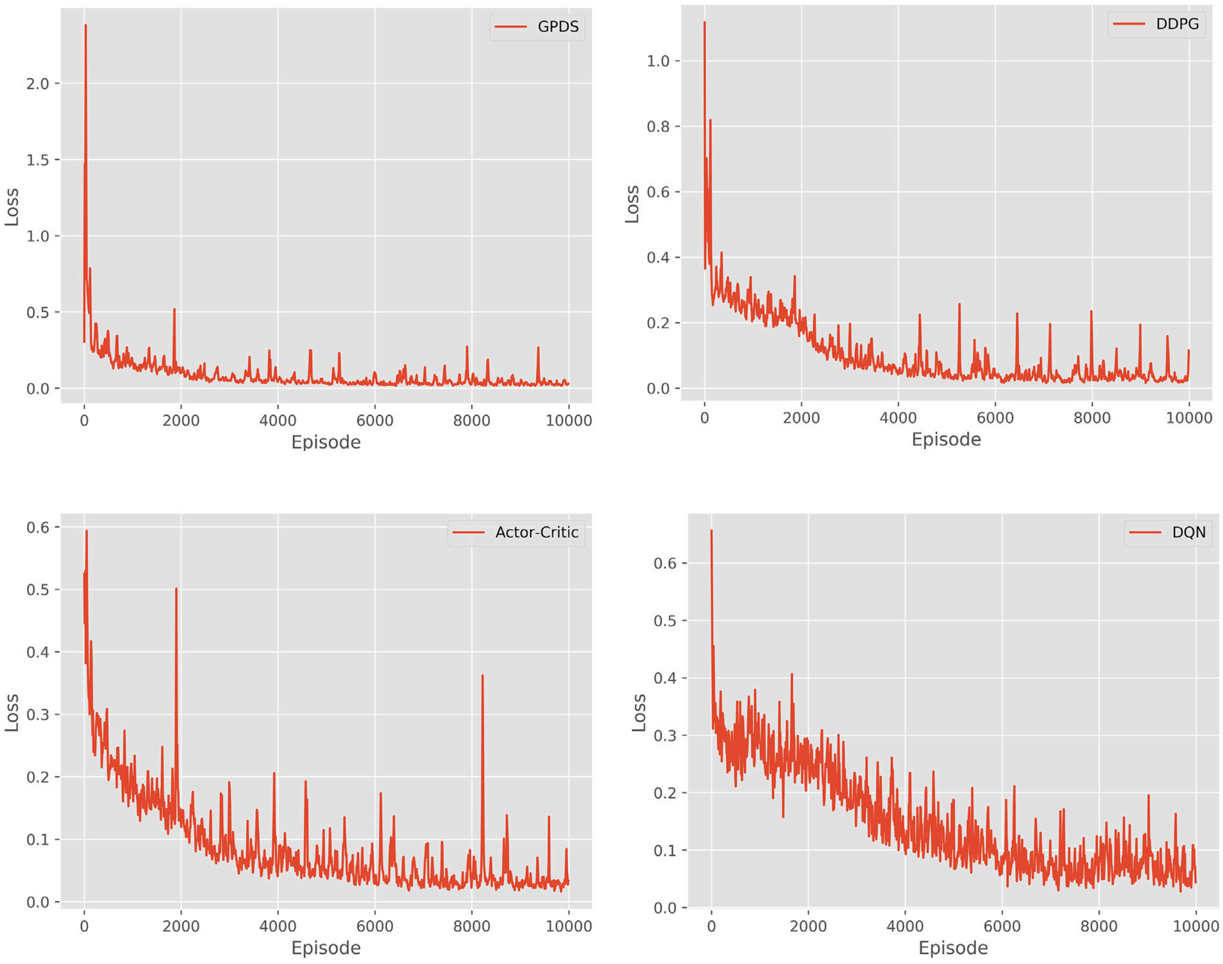


Fig. 6. Comparison of loss after 10 000 episode under GPDS, DDPG, Actor–Critic and DQN algorithms.

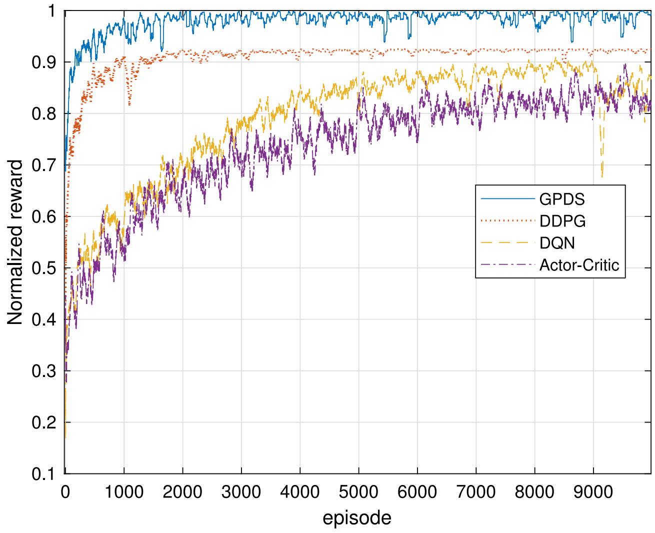


Fig. 7. Normalized reward after 10 000 episodes under GPDS, DDPG, Actor–Critic and DQN algorithms.

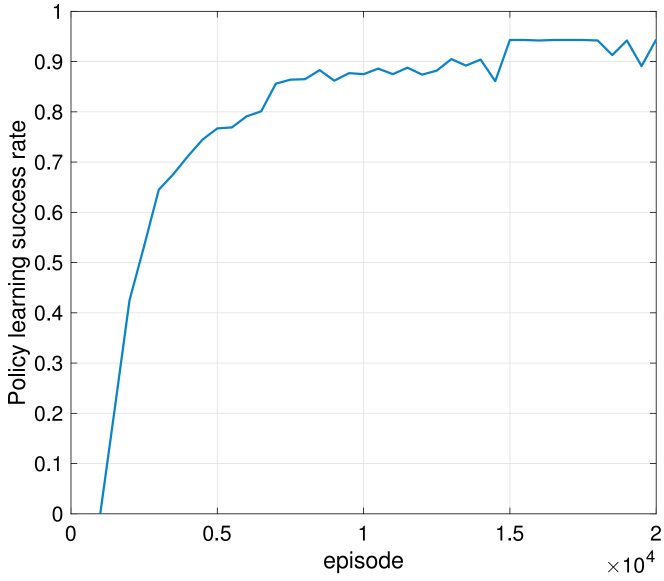


Fig. 8. The policy learning success rate.

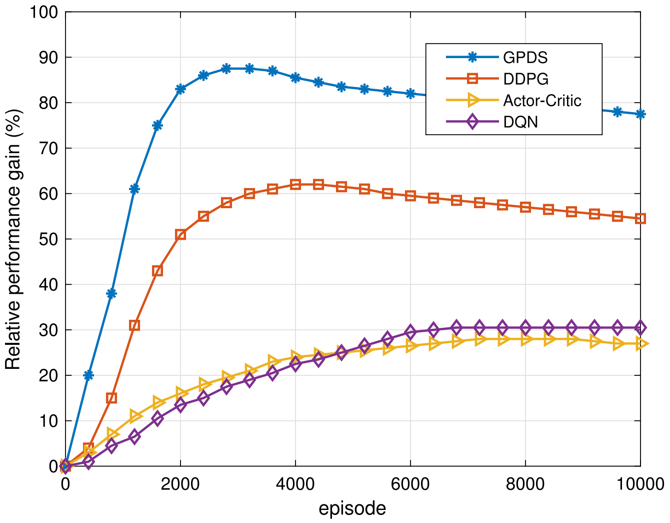


Fig. 9. Relative performance gain $\Phi(n)$ of 10 000 episode under GPDS, DDPG, Actor-Critic and DQN algorithms.

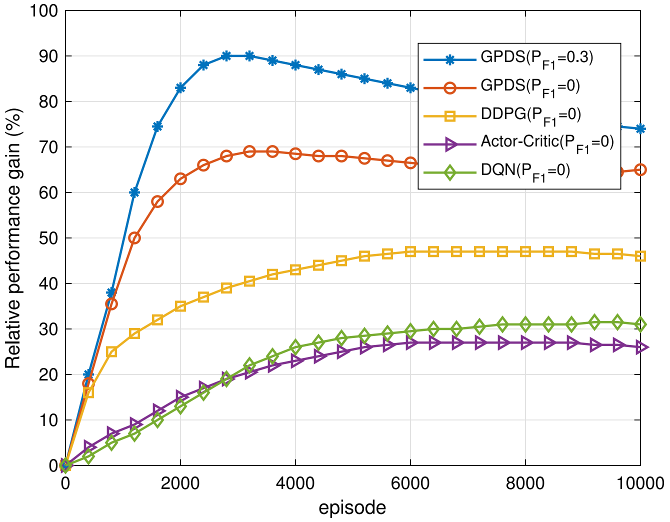


Fig. 10. Relative performance gain $\Phi(n)$ under suspected interference.

can see, using the centralized critic GPDS method are more easily enabled to learn policy behavior. Although the simplicity of the environment, traditional reinforcement learning algorithms, such as DDPG, Actor–Critic, and DQN all fail to learn the correct policy.

For the anti interference system, the $\Phi_{0,1}, \Phi_{1,0}$ of IRS are set to 0.5 and $c_s = 0.5$, $c_p = 2$. Let $k = 2$, the two spectrum channel varies between $h^1 = \{h^{1,1}, h^{1,2}\}$ and $h^2 = \{h^{2,1}, h^{2,2}\}$ with transition probabilities

$$p_{g^1} = \begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix} \text{ and } p_{g^2} = \begin{bmatrix} 0.3 & 0.7 \\ 0.4 & 0.6 \end{bmatrix}.$$

And $C(h^{1,1}) = 1, C(h^{1,2}) = 2, C(h^{2,1}) = 0, C(h^{2,2}) = 1$ with transition matrix

$$p_D = \begin{bmatrix} 0.3 & 0.3 & 0.4 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 1 \end{bmatrix}.$$

Let $p_{FI} = 0.3$ be the probability that the mobile user transmits (Encounter suspected interference). The performance of our proposed GPDS strategy is compared with that of traditional methods is shown in Fig. 10.

To further test the performance of the proposed algorithm, we considered 4 agents in the security game. Their goal is to maximize the expected intrusion benefits. Meanwhile, two defenders try to defend the attackers and minimize the loss caused by being attacked. In the proposed algorithm, defender agent will imitate and learn from the experienced agents. The information involved in the game process is represented by utility matrix and transfer matrix.

Initialize the four original transfer matrices as:

$$\begin{bmatrix} 0.2816 & 0.2354 & 0.2556 & 0.2274 \\ 0.2473 & 0.2546 & 0.2813 & 0.2168 \\ 0.2451 & 0.3212 & 0.1546 & 0.2791 \\ 0.2798 & 0.2121 & 0.2465 & 0.2616 \end{bmatrix},$$

$$\begin{bmatrix} 0.3461 & 0.3549 & 0.1245 & 0.1745 \\ 0.1677 & 0.2364 & 0.3247 & 0.2712 \\ 0.3312 & 0.1846 & 0.2465 & 0.2377 \\ 0.1986 & 0.1876 & 0.3214 & 0.2924 \end{bmatrix},$$

$$\begin{bmatrix} 0.1846 & 0.2346 & 0.2474 & 0.3334 \\ 0.1465 & 0.3461 & 0.1563 & 0.3511 \\ 0.2414 & 0.2341 & 0.1992 & 0.3253 \\ 0.3127 & 0.2416 & 0.2013 & 0.2444 \end{bmatrix},$$

$$\begin{bmatrix} 0.3411 & 0.1774 & 0.3123 & 0.1692 \\ 0.3213 & 0.3001 & 0.2104 & 0.1682 \\ 0.2461 & 0.2366 & 0.3124 & 0.2049 \\ 0.2013 & 0.3002 & 0.2781 & 0.2204 \end{bmatrix}.$$

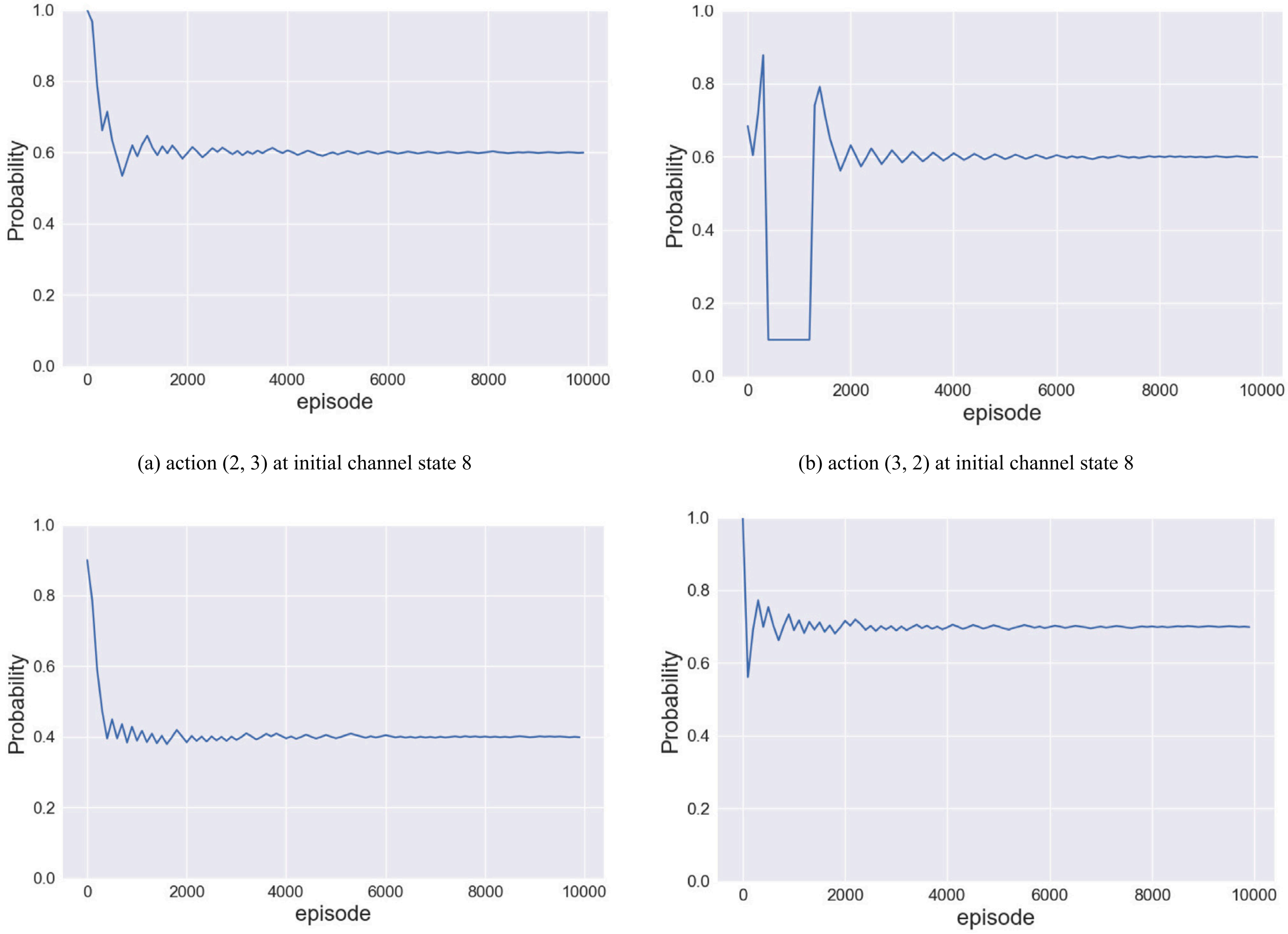
Also, the utility matrices are showed as:

$$\begin{bmatrix} 95 & 142 & 142 & 82 \\ 82 & 165 & 121 & 96 \\ 37 & 210 & 172 & 102 \\ 66 & 155 & 143 & 96 \end{bmatrix}, \begin{bmatrix} 58 & 145 & 135 & 221 \\ 78 & 184 & 211 & 114 \\ 82 & 150 & 180 & 190 \\ 65 & 200 & 142 & 101 \end{bmatrix},$$

$$\begin{bmatrix} 46 & 83 & 94 & 76 \\ 61 & 77 & 121 & 112 \\ 83 & 56 & 93 & 134 \\ 99 & 145 & 76 & 51 \end{bmatrix}, \begin{bmatrix} 35 & 165 & 44 & 59 \\ 57 & 195 & 87 & 87 \\ 49 & 164 & 71 & 130 \\ 90 & 186 & 143 & 76 \end{bmatrix}.$$

Consider the channel gain h^z value from $\{1, 8\}$, $T_t^S \in \{0, 1\}$ indicates whether mobile user can send/sense to be assigned to mobile users, such as, the random spectrum access environment in MEC networks, $P^u(T_t^S|T_t^{on})$ (T_t^{on} means IRS is active) is only known to the mobile users, Θ_t denotes the data channels. If $T_t^S = 0$, the mobile users will be waiting.

At first, assumed that all agents at initial channel state 8, that is $h_1^1 = h_1^2 = h_1^3 = h_1^4 = 8$. The learning cure of all mobile users at state 8 are shown in Fig. 11 and the Learning curve of the attackers is shown in Fig. 12. Form Fig. 11, we can see that, after 2000 episodes, the algorithm will converge to the optimal policy. According to Fig. 11, the SUs mostly take action the action (2, 3) with probability 0.75, action (3, 2) with probability 0.6, action (5, 1) with probability 0.43 and action (5, 2) with probability 0.78; According to Fig. 12, the attackers take action 2 with probability 0.4, action 1 with probability 0.25, action 2 with probability 0.45, action 3 with probability 0.45. The experimental results show that the channel 2 has high availability, so attackers tend to send spoofing signals to attack and attempt to block this channel. However, the attackers still chooses to attack other channels, which indicates that the attacker's strategy is random. Since the attacker's strategy is random, the optimal defense game strategy can bring more benefits to mobile users. Note that the curve in Fig. 11(b) drops sharply and then rises rapidly. The reason is that reinforcement learning is a trial-and-error learning process, and agents need to find good policies in their interactions with the environment. However, in the learning process, random initialization may produce a great deviation value.



(a) action (2, 3) at initial channel state 8

(b) action (3, 2) at initial channel state 8

(c) action (5, 1) at initial channel state 8

(d) action (5, 2) at initial channel state 8

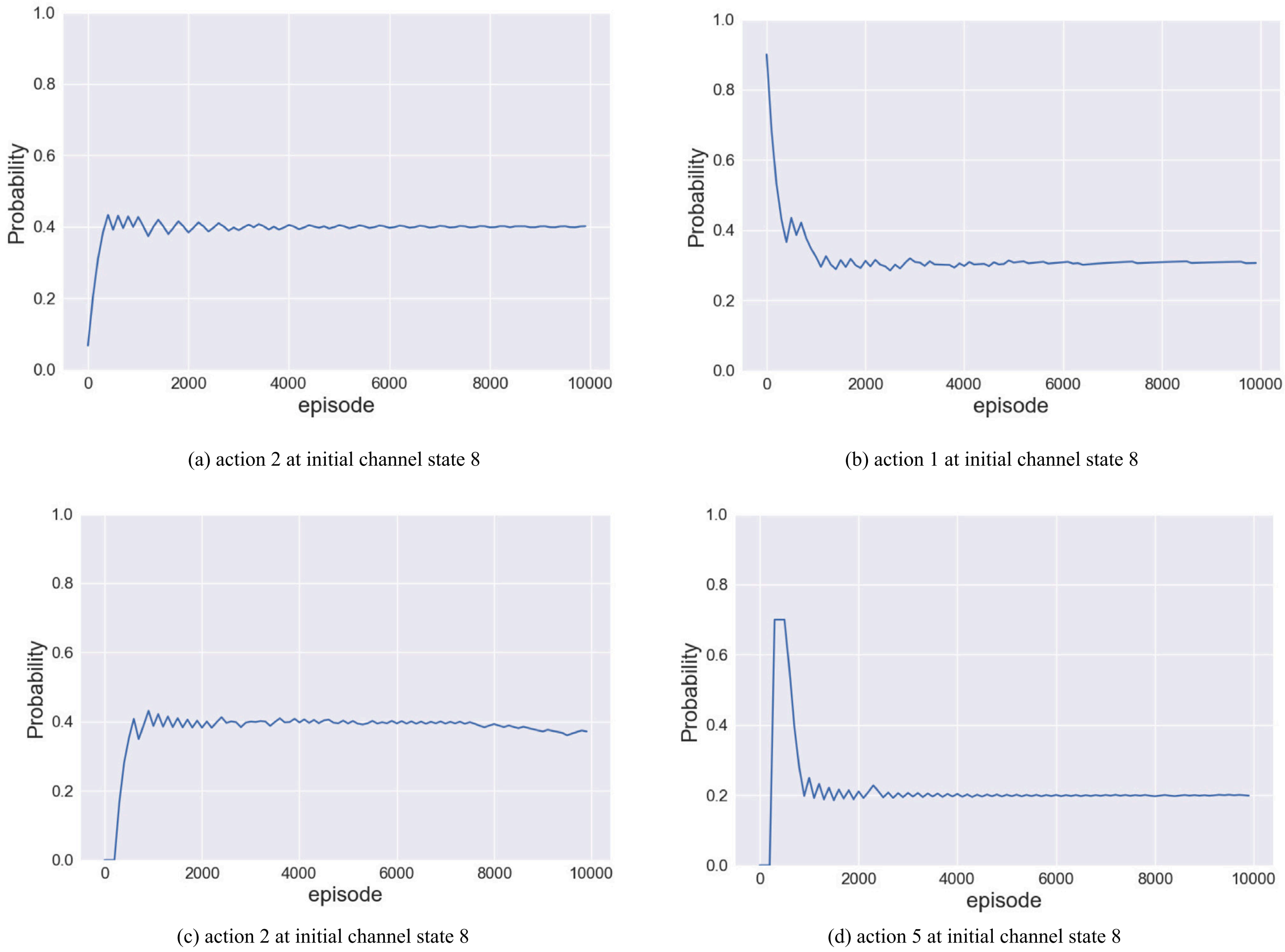
Fig. 11. Learning curve of the mobile users at initial channel state 8.

At this time, the agent needs to maximize reward according to the known information through continuous exploration, constantly correct the error value, and finally converge to the optimal solution.

The transmission rate (Byte/s/Hz) and SINR protection performance comparison for different transmit power (dBm) are shown in Figs. 13 and 14, where mobile users $N = 10$. In Figs. 13(a) and 14(a), we set the number of IRS elements is $L = 80$, obviously, the transmission rate and protection level improve as transmit power P increases. In addition, under different P values, the proposed GPDS learning method, DDPG, and actor–critic algorithms have good channel transmission rate values, and they are much better than the DQN algorithm based on discrete model. However, the channel transmission rate and SINR protection performance of the three SOTA algorithms are significantly lower than that of the proposed GPDS algorithm, because they are a single user optimization solution that ignores multi-user assistance games, which cannot effectively ensure the performance requirements. Figs. 13(b) and 14(b) show the transmission rate (Byte/s/Hz) and SINR protection performance of the 4 algorithms for different IRS elements when $P = 40$ dBm. Because all algorithms are based on IRS deployment strategy, the transmission rate performance of all algorithms increases with the increase of the number of IRS elements. Therefore, the multi-user intelligent game and IRS strategy used in the proposed GPDS algorithm can achieve a higher level of performance gain. According to the experimental results, the transmission power can be increased by increasing the number of IRS elements, the channel communication quality can be improved by the IRS phase, and the intelligent interference of jammer can be reduced. In addition, compared with the other three SOTA algorithms, the proposed GPDS algorithm only needs 80 IRS elements to have sufficient transmission power for anti-interference defense, and can achieve 100% protection level. This is because the proposed GPDS algorithm can achieve more flexible joint power allocation and reflected beam optimization, and improve the performance of intelligent jamming in MEC networks.

6. Conclusion and future work

In this paper, we have investigated the anti jamming and security mechanisms for mobile devices in time-varying MEC networks, under the framework of anti jamming security game and deep learning. The conventional value-based anti jamming security strategies cannot work well to protect the communication security in the considered scenario due to the high electricity resource consumption and algorithm complexity, presence of smart jammer attackers, and dynamics in human–robot interactive environments. To solve this problem, we proposed a novel policy-based multi-agent deep reinforcement learning algorithm, i.e., GPDS, which can improve the SINR protection level and channel throughput of mobile devices in unknown and dynamic communication jamming environments. In order to realize multi-agent game, we combine minimax strategy with multi-user deep reinforcement learning, and use user to user multi-user adaptive learning for training to realize nonlinear processing of continuous channel variables. Through simulations, it is demonstrated that the proposed learning algorithm



(a) action 2 at initial channel state 8

(b) action 1 at initial channel state 8

(c) action 2 at initial channel state 8

(d) action 5 at initial channel state 8

Fig. 12. Learning curve of the attackers at initial channel state 8.

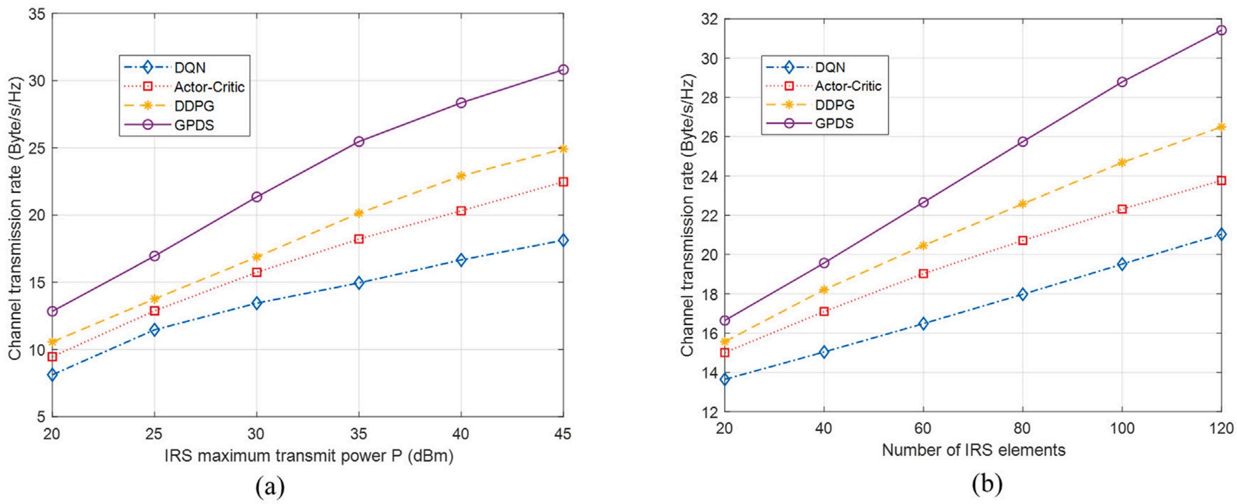


Fig. 13. The transmission rate (Byte/s/Hz) comparisons versus the maximum transmit power and the number of IRS elements under GPDS, DDPG, Actor–Critic and DQN algorithms.

outperforms traditional SOTA reinforcement learning algorithms in a multi-agent anti jamming environment. For the future work, we will try to accelerate the speed of GPDS algorithm to adapt to fast-changing dynamic security game environments.

CRediT authorship contribution statement

Miaojiang Chen: Data curation, Software, Visualization, Writing – original draft. **Wei Liu:** Conceptualization, Methodology, Funding

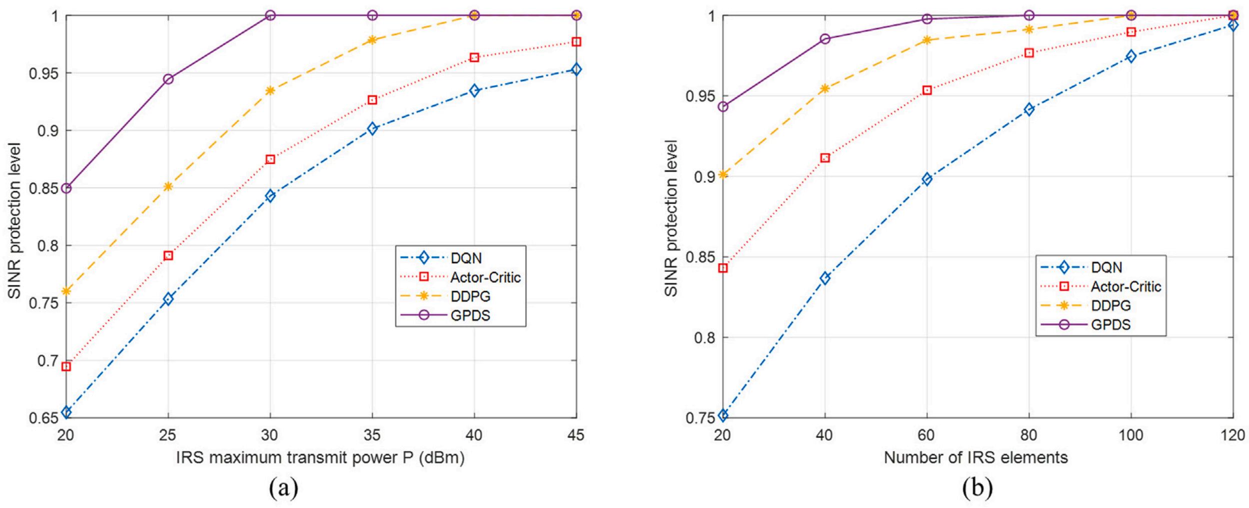


Fig. 14. SINR protection level comparisons versus the maximum transmit power and the number of IRS elements under GPDS, DDPG, Actor–Critic and DQN algorithms.

acquisition, Validation, Project administration, Resources. **Ning Zhang:** Supervision, Writing – review & editing. **Junling Li:** Supervision, Writing – review & editing. **Yingying Ren:** Formal analysis. **Meng Yi:** Data curation, Software, Visualization, Writing – original draft. **Anfeng Liu:** Conceptualization, Methodology, Funding acquisition, Validation, Project administration, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62072475, 61772554), Supported by the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No. ICT2022B15), part by Hunan Provincial Innovation Foundation for Postgraduate, China (Project Leader: Miaojiang Chen). We are very grateful to the anonymous reviewers for their constructive comments that were helpful for us to improve the manuscript.

References

Chen, G., & Dong, W. (2020). Reactive jamming and attack mitigation over cross-technology communication links. *ACM Transactions on Sensor Networks*, *17*(1), 1–25.

Chen, M., Liu, A., Liu, W., Ota, K., Dong, M., & Xiong, N. N. (2021). RDRL: A recurrent deep reinforcement learning scheme for dynamic spectrum access in reconfigurable wireless networks. *IEEE Transactions on Network Science and Engineering*, *9*(2), 364–376.

Chen, M., Liu, W., Wang, T., Liu, A., & Zeng, Z. (2021). Edge intelligence computing for mobile augmented reality with deep reinforcement learning approach. *Computer Networks*, Article 108186.

Chen, M., Liu, W., Wang, T., Zhang, S., & Liu, A. (2021). A game-based deep reinforcement learning approach for energy-efficient computation in MEC systems. *Knowledge-Based Systems*, Article 107660.

Chen, C., Song, M., Xin, C., & Backens, J. (2013). A game-theoretical anti-jamming scheme for cognitive radio networks. *IEEE Network*, *27*(3), 22–27.

Chen, M., Wang, T., Zhang, S., & Liu, A. (2021). Deep reinforcement learning for computation offloading in mobile edge computing environment. *Computer Communications*, *175*, 1–12.

Di Renzo, M., Zappone, A., Debbah, M., Alouini, M.-S., Yuen, C., De Rosny, J., et al. (2020). Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead. *IEEE Journal on Selected Areas in Communications*, *38*(11), 2450–2525.

Do, T. T., Björnson, E., Larsson, E. G., & Razavizadeh, S. M. (2018). Jamming-resistant receivers for the massive MIMO uplink. *IEEE Transactions on Information Forensics and Security*, *13*(1), 210–223. http://dx.doi.org/10.1109/TIFS.2017.2746007.

D'Oro, S., Ekici, E., & Palazzo, S. (2018). Optimal power allocation and scheduling under jamming attacks. *IEEE/ACM Transactions on Networking*, *25*(3), 1310–1323. http://dx.doi.org/10.1109/TNET.2016.2622002.

El-Bardan, R., Brahma, S., & Varshney, P. K. (2016). Strategic power allocation with incomplete information in the presence of a jammer. *IEEE Transactions on Communications*, *64*(8), 3467–3479. http://dx.doi.org/10.1109/TCOMM.2016.2577686.

Elleuch, I., Pourranjbar, A., & Kaddoum, G. (2021). A novel distributed multi-agent reinforcement learning algorithm against jamming attacks. *IEEE Communications Letters*, *25*(10), 3204–3208.

Fan, H., Yang, Z., Zhang, X., Wu, S., Long, J., & Liu, L. (2022). A novel multi-satellite and multi-task scheduling method based on task network graph aggregation. *Expert Systems with Applications*, Article 117565.

Feng, S., & Haykin, S. (2019). Cognitive risk control for anti-jamming V2V communications in autonomous vehicle networks. *IEEE Transactions on Vehicular Technology*, *68*(10), 9920–9934.

Gao, N., Qin, Z., Jing, X., Ni, Q., & Jin, S. (2019). Anti-intelligent UAV jamming strategy via deep Q-networks. *IEEE Transactions on Communications*, *68*(1), 569–581.

Gao, Y., Xiao, Y., Wu, M., Xiao, M., & Shao, J. (2018). Game theory-based anti-jamming strategies for frequency hopping wireless communications. *IEEE Transactions on Wireless Communication*, *17*(8), 5314–5326. http://dx.doi.org/10.1109/TWC.2018.2841921.

Gu, P., Hua, C., Xu, W., Khatoun, R., Wu, Y., & Serhrouchni, A. (2020). Control channel anti-jamming in vehicular networks via cooperative relay beamforming. *IEEE Internet of Things Journal*, *7*(6), 5064–5077. http://dx.doi.org/10.1109/JIOT.2020.2973753.

Hanawal, M. K., Abdel-Rahman, M. J., & Krunz, M. (2017). Joint adaptation of frequency hopping and transmission rate for anti-jamming wireless systems. *IEEE Transactions on Mobile Computing*, *15*(9), 2247–2259. http://dx.doi.org/10.1109/TMC.2015.2492556.

Hoang, T. M., Duong, T. Q., Suraweera, H. A., Tellambura, C., & Poor, H. V. (2015). Cooperative beamforming and user selection for improving the security of relay-aided systems. *IEEE Transactions on Communications*, *63*(12), 5039–5051.

Hu, J., Wellman, M. P., et al. (1998). Multiagent reinforcement learning: theoretical framework and an algorithm. In *ICML, Vol. 98* (pp. 242–250). Citeseer.

Huang, M., Leung, V. C., Liu, A., & Xiong, N. N. (2022). TMA-DPSO: Towards efficient multi-task allocation with time constraints for next generation multiple access. *IEEE Journal on Selected Areas in Communications*, *40*(5), 1652–1666.

Huang, C., Zappone, A., Alexandropoulos, G. C., Debbah, M., & Yuen, C. (2019). Reconfigurable intelligent surfaces for energy efficiency in wireless communication. *IEEE Transactions on Wireless Communication*, *18*(8), 4157–4170. http://dx.doi.org/10.1109/TWC.2019.2922609.

Huynh, H. A., Han, Y., Park, S., Hwang, J., Song, E., & Kim, S. (2018). Design and analysis of the DC–DC converter with a frequency hopping technique for EMI reduction. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, *8*(4), 546–553.

Jia, L., Xu, Y., Sun, Y., Feng, S., Yu, L., & Anpalagan, A. (2018). A game-theoretic learning approach for anti-jamming dynamic spectrum access in dense wireless networks. *IEEE Transactions on Vehicular Technology*, *68*(2), 1646–1656.

Law, Y. W., Palaniswami, M., Hoesel, L. V., Doumen, J., Hartel, P., & Havinga, P. (2009). Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols. *ACM Transactions on Sensor Networks*, *5*(1), 1–38.

Lee, S., Kim, S., Seo, M., & Har, D. (2019). Synchronization of frequency hopping by LSTM network for satellite communication system. *IEEE Communications Letters*, *23*(11), 2054–2058.

Lee, A. X., Nagabandi, A., Abbeel, P., & Levine, S. (2020). Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, *33*, 741–752.

Li, J., Li, T., Liu, Z., & Chen, X. (2019). Secure deduplication system with active key update and its application in IoT. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *10*(6), 1–21.

Liang, L., Cheng, W., Zhang, W., & Zhang, H. (2018). Mode hopping for anti-jamming in radio vortex wireless communications. *IEEE Transactions on Vehicular Technology*, *67*(8), 7018–7032. http://dx.doi.org/10.1109/TVT.2018.2825539.

Liu, Q., Li, M., Kong, X., & Zhao, N. (2016). Disrupting MIMO communications with optimal jamming signal design. *IEEE Transactions on Wireless Communication*, *14*(10), 5313–5325. http://dx.doi.org/10.1109/TWC.2015.2436385.

Lv, S., Xiao, L., Hu, Q., Wang, X., Hu, C., & Sun, L. (2017). Anti-jamming power control game in unmanned aerial vehicle networks. In *GLOBECOM 2017-2017 IEEE global communications conference* (pp. 1–6). IEEE.

Mukherjee, A., Fakoorian, S. A. A., Huang, J., & Swindlehurst, A. L. (2014). Principles of physical layer security in multiuser wireless networks: A survey. *IEEE Communications Surveys & Tutorials*, *16*(3), 1550–1573.

Ren, Y., Liu, W., Liu, A., Wang, T., & Li, A. (2022). A privacy-protected intelligent crowdsourcing application of IoT based on the reinforcement learning. *Future Generation Computer Systems*, *127*, 56–69.

Rowland, M., Omidshafiei, S., Tuyls, K., Perolat, J., Valko, M., Piliouras, G., et al. (2019). Multiagent evaluation under incomplete information. *Advances in Neural Information Processing Systems*, *32*.

Shi, Y., An, K., & Li, Y. (2021). Index modulation based frequency hopping: Anti-jamming design and analysis. *IEEE Transactions on Vehicular Technology*, *70*(7), 6930–6942. http://dx.doi.org/10.1109/TVT.2021.3087640.

Tang, X., Wang, D., Zhang, R., Chu, Z., & Han, Z. (2021). Jamming mitigation via aerial reconfigurable intelligent surface: Passive beamforming and deployment optimization. *IEEE Transactions on Vehicular Technology*, *70*(6), 6232–6237. http://dx.doi.org/10.1109/TVT.2021.3077662.

Torreño, A., Onaindia, E., & Sapena, O. (2015). An approach to multi-agent planning with incomplete information. arXiv preprint arXiv:1501.07256.

Van Huynh, N., Nguyen, D. N., Hoang, D. T., & Dutkiewicz, E. (2019). “Jam me if you can:” Defeating jammer with deep dueling neural network architecture and ambient backscattering augmented communications. *IEEE Journal on Selected Areas in Communications*, *37*(11), 2603–2620. http://dx.doi.org/10.1109/JSAC.2019.2933889.

Van Huynh, N., Nguyen, D. N., Hoang, D. T., & Dutkiewicz, E. (2020). “Jam me if you can:” defeating jammer with deep dueling neural network architecture and ambient backscattering augmented communications. *IEEE Journal on Selected Areas in Communications*, *37*(11), 2603–2620.

Wang, C., & Wang, H.-M. (2015). Robust joint beamforming and jamming for secure AF networks: Low-complexity design. *IEEE Transactions on Vehicular Technology*, *64*(5), 2192–2198. http://dx.doi.org/10.1109/TVT.2014.2334640.

Wang, X., Wang, J., Xu, Y., Chen, J., Jia, L., Liu, X., et al. (2020). Dynamic spectrum anti-jamming communications: Challenges and opportunities. *IEEE Communications Magazine*, *58*(2), 79–85.

Wu, Q., & Zhang, R. (2019). Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming. *IEEE Transactions on Wireless Communication*, *18*(11), 5394–5409.

Xiao, L., Li, Y., Dai, C., Dai, H., & Poor, H. V. (2018). Reinforcement learning-based NOMA power allocation in the presence of smart jamming. *IEEE Transactions on Vehicular Technology*, *67*(4), 3377–3389. http://dx.doi.org/10.1109/TVT.2017.2782726.

Xiao, L., Liu, J., Li, Q., Mandayam, N. B., & Poor, H. V. (2015). User-centric view of jamming games in cognitive radio networks. *IEEE Transactions on Information Forensics and Security*, *10*(12), 2578–2590.

Xiao, L., Lu, X., Xu, D., Tang, Y., Wang, L., & Zhuang, W. (2018b). UAV relay in VANETs against smart jamming with reinforcement learning. *IEEE Transactions on Vehicular Technology*, *67*(5), 4087–4097.

Xiao, L., Lu, X., Xu, D., Tang, Y., Wang, L., & Zhuang, W. (2018c). UAV relay in VANETs against smart jamming with reinforcement learning. *IEEE Transactions on Vehicular Technology*, *67*(5), 4087–4097. http://dx.doi.org/10.1109/TVT.2018.2789466.

Xiong, Z., Zhang, Y., Lim, W. Y. B., Kang, J., Niyato, D., Leung, C., et al. (2020). UAV-assisted wireless energy and data transfer with deep reinforcement learning. *IEEE Transactions on Cognitive Communications and Networking*, *7*(1), 85–99.

Xiong, Z., Zhang, Y., Niyato, D., Deng, R., Wang, P., & Wang, L.-C. (2019). Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges. *IEEE Vehicular Technology Magazine*, *14*(2), 44–52.

Xiong, Z., Zhao, J., Niyato, D., Deng, R., & Zhang, J. (2020). Reward optimization for content providers with mobile data subsidization: A hierarchical game approach. *IEEE Transactions on Network Science and Engineering*, *7*(4), 2363–2377.

Xu, D. (2020). Proactive eavesdropping of suspicious non-orthogonal multiple access networks. *IEEE Transactions on Vehicular Technology*, *69*(11), 13958–13963. http://dx.doi.org/10.1109/TVT.2020.3021953.

Xu, D., & Zhu, H. (2022a). Jamming-assisted legitimate eavesdropping and secure communication in multicarrier interference networks. *IEEE Systems Journal*, *16*(1), 954–965. http://dx.doi.org/10.1109/JSYST.2020.3030574.

Xu, D., & Zhu, H. (2022b). Legitimate surveillance of suspicious computation offloading in mobile edge computing networks. *IEEE Transactions on Communications*, *70*(4), 2648–2662. http://dx.doi.org/10.1109/TCOMM.2022.3151767.

Yang, H., Xiong, Z., Zhao, J., Niyato, D., Wu, Q., Tornatore, M., et al. (2020). Intelligent reflecting surface assisted anti-jamming communications based on reinforcement learning. In *GLOBECOM 2020-2020 IEEE global communications conference* (pp. 1–6). IEEE.

Yao, F., & Jia, L. (2019). A collaborative multi-agent reinforcement learning anti-jamming algorithm in wireless networks. *IEEE Wireless Communications Letters*, *8*(4), 1024–1027. http://dx.doi.org/10.1109/LWC.2019.2904486.

Yin, Z., Lin, Y., Zhang, Y., Qian, Y., Shu, F., & Li, J. (2022). Collaborative multi-agent reinforcement learning aided resource allocation for UAV anti-jamming communication. *IEEE Internet of Things Journal*, 1. http://dx.doi.org/10.1109/JIOT.2022.3188833.

Yu, M., Liu, A., Xiong, N. N., & Wang, T. (2022). An intelligent game-based offloading scheme for maximizing benefits of IoT-edge-cloud ecosystems. *IEEE Internet of Things Journal*, *9*(8), 5600–5616. http://dx.doi.org/10.1109/JIOT.2020.3039828.

Yu, Y., Tang, J., Huang, J., Zhang, X., So, D. K. C., & Wong, K.-K. (2021). Multi-objective optimization for UAV-assisted wireless powered IoT networks based on extended DDPG algorithm. *IEEE Transactions on Communications*, *69*(9), 6361–6374.

Zheng, X., & Cai, Z. (2020). Privacy-preserved data sharing towards multiple parties in industrial IoTs. *IEEE Journal on Selected Areas in Communications*, *38*(5), 968–979. http://dx.doi.org/10.1109/JSAC.2020.2980802.

Zhu, S., Li, W., Li, H., Tian, L., Luo, G., & Cai, Z. (2018). Coin hopping attack in blockchain-based IoT. *IEEE Internet of Things Journal*, *6*(3), 4614–4626.

Miaojiang Chen received the B.S. degree in computer science from Guangxi University in 2018. He is currently a Ph.D. candidate with School of Computer Science and Engineering of Central South University, China. He has published several journal and conference papers in the IEEE transactions on network science and engineering, Knowledge-Based Systems, Computer Network, etc., and he also serves reviewer of the top-tier conferences and journals, e.g., International Conference on Machine Learning (ICML), IEEE transactions on industrial informatics, Knowledge-Based Systems. His major research interests include deep reinforcement learning, Internet of Things, edge computing, neural network optimization.

Wei Liu is an associate professor and senior engineer at the School of Informatics, Hunan University of Chinese Medicine, China. He received his Ph.D. degree in computer application technology from Central South University, 2014, China. His research interests include software engineering, data mining and medical informatics. He has published over 20 papers in the related fields.

Ning Zhang (Senior Member, IEEE) is an Associate Professor in the Department of Electrical and Computer Engineering at University of Windsor, Canada. He received the Ph.D. degree in Electrical and Computer Engineering from University of Waterloo, Canada, in 2015. After that, he was a postdoc research fellow at University of Waterloo and University of Toronto, Canada, respectively. His research interests include connected vehicles, mobile edge computing, wireless networking, and machine learning. He is a Highly Cited Researcher. He received an NSERC PDF award in 2015 and 6 Best Paper Awards from IEEE Globecom in 2014, IEEE WCSP in 2015, IEEE ICC in

2019, IEEE ICCC in 2019, IEEE Technical Committee on Transmission Access and Optical Systems in 2019, and Journal of Communications and Information Networks in 2018, respectively. He serves as an Associate Editor of IEEE Internet of Things Journal, IEEE Transactions on Cognitive Communications and Networking, and IEEE Systems Journal; and a Guest Editor of several international journals, such as IEEE Wireless Communications, IEEE Transactions on Industrial Informatics, IEEE Transactions on Intelligent Transportation Systems, and IEEE Transactions on Cognitive Communications and Networking. He also serves/served as a general chair for IEEE SAGC 2021, TPC chair for IEEE SAGC 2020, a track chair for several international conferences including IEEE ICC 2022, CollaborateCom 2021, IEEE VTC 2020, AICON 2020 and CollaborateCom 2020, and a co-chair for numerous international workshops.

Junling Li (IEEE S'18) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada in 2020. She is currently a Joint Postdoctoral Research Fellow at Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), the Chinese University of Hong Kong, Shenzhen, and University of Waterloo. She received the M.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2016, and the B.S. degree from Tianjin University, Tianjin, China, in 2013. Her interests include game theory, machine learning, software-defined networking, network function virtualization, and vehicular networks. She received the Best Paper Award at the IEEE/CIC International Conference on Communications in China (ICCC) in 2019.

Yingying Ren Currently is a master student in School of Information Science and Engineering, Central South University, China. Her research interests include services-based network, crowd sensing networks, and wireless sensor networks.

Meng Yi received the master degree in computer science from Guangxi University in 2020. She is currently a Ph.D. candidate with School of Computer Science and Engineering, Southeast University, China. Her major research interests include deep reinforcement learning, Internet of Things, multiaccess edge computing, heterogeneous wireless networks.

Anfeng Liu received the M.Sc. and Ph.D. degrees from Central South University, China, in 2002 and 2005, respectively, both in computer science. He is currently a professor of the School of Information Science and Engineering, Central South University, China. His major research interest is wireless sensor networks, Internet of Things, information security, edge computing and crowdsourcing. Dr. Liu has published 4 books and over 100 international journal and conference papers, among which there are more than 30 ESI highly-cited papers. Some of his works were published in IEEE Transactions on Information Forensics & Security, IEEE Transactions on Mobile Computing, IEEE Transactions on Services Computing, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Vehicular Technology, IEEE Transactions on ComputerAided Design of Integrated Circuits and Systems, IEEE System Journal, IEEE Wireless Communications, IEEE Communications Magazine, IEEE Network Magazine, IEEE Transactions on Industrial Informatics, ACM Transactions on Embedded Computing Systems, IEEE Internet of Things Journal, IEEE Transactions on Emerging Topics in Computing, IEEE Transactions on Systems Man Cybernetics-Systems. His research has been supported by the National Basic Research Program of China (973 Program) and the National Natural Science Foundation of China for five times. He was a recipient of the First Prize of Scientific Research Achievement of Colleges from the Ministry of Education of China in 2016, and the Second Prize of Science and Technology Award from China Nonferrous Metal Industry Association in 2005. He has served as the Leading Editor of the special issue for International Journal of Distributed Sensor Networks, and a guest editor of the special issues for Scientific Programming, and Sensors. He also serves as a reviewer of over 30 international academic journals.