Received 6 January 2021; accepted 25 January 2021. Date of publication 2 February 2021; date of current version 13 April 2021.

Digital Object Identifier 10.1109/OJCOMS.2021.3056113

Intelligent Anti-Jamming Communication for Wireless Sensor Networks: A Multi-Agent Reinforcement Learning Approach

QUAN ZHOU [1], YONGGUI LI [2], AND YINGTAO NIU [2]

[1]College of Communications Engineering, Army Engineering University of PLA, Nanjing 210007, China

[2]The Sixty-Third Research Institute, National University of Defense Technology, Nanjing 410073, China

CORRESPONDING AUTHOR: Y. NIU (e-mail: niuyingtao78@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant U19B2014; and in part by the Foundation Strengthening Program Area Fund of China under Grant 2019-JCJQ-JJ-212.

ABSTRACT In this article, we investigate intelligent anti-jamming communication method for wireless sensor networks. The stochastic game framework is introduced to model and analyze the multi-user anti-jamming problem, and a joint multi-agent anti-jamming algorithm (JMAA) is proposed to obtain the optimal anti-jamming strategy. In intelligent multi-channel blocking jamming environment, the proposed JMAA adopts multi-agent reinforcement learning to make online channel selection, which can effectively tackle the external malicious jamming and avoid the internal mutual interference among sensor nodes. The simulation results show that, the proposed JMAA is superior to the frequency-hopping method, the sensing-based method and the independent reinforcement learning. Specifically, the proposed JMAA has the higher average packet receive ratio than both the frequency-hopping method and the sensing-based method. Compared with the independent reinforcement learning, JMAA has faster convergence rate when reaching the same performance of average packet receive ratio. In addition, since the JMAA does not need to model the jamming patterns, it can be widely used for combating other malicious jamming such as sweep jamming and probabilistic jamming.

INDEX TERMS Communication anti-jamming, channel selection, multi-agent reinforcement learning, Q-learning, wireless sensor networks.

I. INTRODUCTION

AS A NOVEL network to realize the comprehensive information interaction between human and the objective world, the Internet of Things (IOT) is based on information perception, transmission and processing. Wireless sensor network (WSN) is an important underlying network technology to realize the wide application of the IOT. It is a short-distance wireless communication network composed of a large number of low-cost, low-power, multi-functional sensor nodes [1], [2]. In recent years, the WSN has engulfed many application fields for its potential advantages [3], [4], [5]. When WSN is applied to some pivotal scenarios, such as traffic monitoring [6], health monitoring [7], military target tracking [8], etc., its information transmission needs to be guaranteed with strict reliability. However, the open transmission medium, the limited computing, storage, and power resources, and the simple network architecture make WSN extensively vulnerable to artificial malicious jamming. For example, a malicious jammer can inject electromagnetic jamming with a certain power into the communication channels to suppress data communication between sensors [9]. Furthermore, a malicious jammer can intercept and thereafter tamper with the data being transmitted, or even masquerade as a sensor node and transmit false data [10]. How to effectively combat various malicious jamming is a significant challenge for WSN.

Spectrum spreading technology is the mainstream communication anti-jamming technology, among which frequency-hopping spectrum spreading (FHSS) [11] and direct sequence spectrum spreading (DSSS) [12] have been widely used.

These technologies have remarkable anti-jamming effect on the conventional jamming, such as sweep jamming, pulse jamming and wideband blocking jamming. However, on the one hand, the traditional anti-jamming methods have some limitations. For example, FHSS relies on a predetermined frequency-hopping pattern [13]. DSSS relies on a local pseudo-random code [14]. On the other hand, with the development of artificial intelligence (AI) and software defined radio (SDR) technology, new trends such as diversity, dynamics and intelligence of jammers [15], [16] have put forward higher requirements for communication anti-jamming technology [17].

In recent years, the machine learning (ML) has been widely concerned in various fields, including wireless communication network. As a popular machine learning technique, deep learning has been widely researched in wireless communication scenarios. In [18], the deep learning algorithm was used to predict the future traffic load and congestion of the IOT, and channel allocation was carried out. In [19], a novel deep learning based algorithm was proposed to realize intelligent traffic control in large scale dynamic network. Reinforcement learning (RL) is another major branch of ML, which mainly focuses on how agents can take different actions in their environment to obtain the maximum reward. The RL represented by Q-learning has been widely researched in wireless communication scenarios. In [20], a Q-learning based algorithm was proposed to learn the optimal channel assignment policy in the mobile communication system. In [21], a Q-learning based approach was proposed to avoid the interference between different cells in the Self-Organized Femtocell Network. Deep reinforcement learning (DRL) is a combination of DL and RL, which is also widely concerned in the field of wireless communication network. In [22], a novel deep reinforcement learning based algorithm was proposed to dynamically allocate radio resources in an online manner for high mobility wireless heterogeneous network. In [23], to maximize the network performance, a deep reinforcement learning based distributed cooperation framework was proposed that allows nodes to assess network conditions and make decisions on whether to keep data communications, defend the network against jamming, or jam other transmissions.

The novel technology mentioned above has pointed out new research directions for both jamming attack and counter malicious jamming. In [24], a deep learning-based jammer is proposed to predict and jam the wireless transmissions. In [25], two different jammers were proposed, namely a feed-forward neural network (FNN) jammer and a deep reinforcement learning (DRL) jammer, to perform the jamming attacks on a user performing dynamic multichannel access using a DRL agent itself. Therefore, based on the idea of "using intelligence to counter intelligence", more efficient intelligent anti-jamming methods are needed.

Machine learning can be divided into supervised learning (SL), unsupervised learning (UL) and reinforcement learning (RL), Unlike SL and UL, RL does not require pre-calibrated data sets for training, and its learning process is characterized by autonomous exploration of optimal strategies. It means that the online learning can be realized by RL. In the communication anti-jamming problem, the jamming environment may change rapidly. Malicious jamming may be dynamic jamming, unknown type jamming or even intelligent jamming, which means that it is difficult to give the training data set in advance. RL in the face of unknown jamming environment can learn the jamming pattern in real time and gradually improve the transmission strategy, which is of great benefit to realize reliable communication in complex dynamic jamming environment. By introducing RL into the anti-jamming problems, users can continuously adjust the transmission strategy by trying different actions under the jamming environment, and finally obtain the optimal strategy. RL such as the classical Q-learning algorithm has been widely used in solving anti-jamming problems [26], [27], [28]. Nevertheless, the existing anti-jamming schemes based on RL also have some limitations and shortcomings. For example, in [29], [30], the optimal frequency-hopping strategy under dynamic jamming environment was obtained by using standard Q-learning or improved Q-learning algorithm. However, only the single-user scenario was considered, and hence it is not applicable to WSN with a large number of sensor nodes. In [31], the anti-jamming problem was extended to multi-user scenario. Each user adopted an independent Q-learning algorithm to obtain the optimal channel switching strategy. Then, the authors in [32] considered the coordination among users, and a collaborative multi-agent anti-jamming algorithm based on RL was proposed to obtain the optimal anti-jamming strategy. However, only the conventional sweep jamming was considered in [31], [32]. The authors of [33] further studied the problem of anti-jamming communication under intelligent comb jamming environment. The deep reinforcement learning (DRL) technology combining deep learning (DL) and RL was introduced to obtain the optimal anti-jamming strategy. However, only the single-user scenario was considered, and the rigorous demand of DRL technology on computing resources limits its application in WSN.

To solve these problems mentioned above, this article investigates the anti-jamming problem of multi-sensor nodes based on multi-agent reinforcement learning (MARL) for the malicious jamming with a certain intelligence, so as to provide a preliminary solution and technical support for the realization of "using intelligence to counter intelligence" in WSNs. Note that the limited transmission distance of sensor nodes makes the compact WSN easy to be covered by high-power jammers. Thus, in order to focus on the problem of MARL-based approach against intelligent dynamic jamming, we can consider the external jamming faced by each node as equivalent, without considering the differences of jamming power or jamming channel faced by different nodes. Specifically, the stochastic game framework is introduced to model and analyze the multi-user anti-jamming problem. Than, to effectively counter the external malicious jamming and avoid the co-channel interference among sensor nodes,

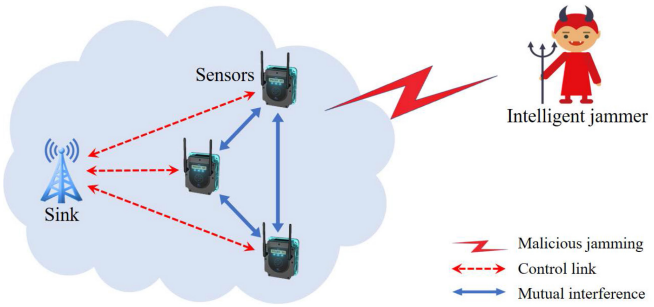


FIGURE 1. System model.

the cooperative learning is considered. Thus, a joint multi-agent anti-jamming algorithm (JMAA) based on multi-agent Q-learning is proposed. The main contributions of this article are as follows:

- In order to avoid external multi-channel intelligent blocking jamming and mutual interference among sensor nodes in WSN, a joint multi-agent anti-jamming algorithm (JMAA) based on multi-agent Q-learning is proposed. The proposed algorithm has the characteristics of "cooperative learning, distributed computing, and centralized decision-making", which can quickly converge to the optimal anti-jamming strategy.
- The proposed algorithm does not need to estimate the jamming patterns or any parameters of the jammer, which can be applied to a variety of anti-jamming scenarios.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. SYSTEM MODEL

The system model is shown in Fig. 1. In order to facilitate the research, we make the following assumptions:

1) the WSN is composed of N sensor nodes and one sink node. The sensor nodes can communicate directly with each other, and the sink node is responsible for coordinating the transmission channel between each sensor node. The set of sensor nodes is denoted as $\mathcal{N} = \{1, \ldots, N\}$. There are M channels in the area that can be used for transmission between sensor nodes. The sensor node does not have a priori knowledge about the channel occupied by other nodes or the jammer, but can sense whether there is external jamming in all M channels. In addition, the sink node and sensor node can achieve reliable signaling interaction through the protocol-reinforced low-capacity control link.
2) The communication/jamming time is divided into communication/jamming timeslots with equal length, which is the minimum time unit for channel switching of the node/jammer. The sensor node divides the communication timeslot into sensing sub-slot, transmission sub-slot and local learning sub-slot. Each transmission sub-slot can transmit one data packet, and an ACK message can be received if the transmission is successful. The sensing sub-slot and local learning sub-slot are used to jamming sensing and local learning respectively. Besides, the sink node divides the communication timeslot into decision-making sub-slot and learning sub-slot. The decision-making sub-slot is used to decide and coordinate the transmission channel of each sensor node. And the learning sub-slot is used to execute the learning algorithm.
3) The high-power jamming signal emitted by an external intelligent jammer can completely cover all sensor nodes. Since the sensor nodes are close to each other, they can be considered to face the same external malicious jamming. Besides, The intelligent jammer can continuously sense all available channel, and the K ($K < M$) channels that are occupied for the longest time in the current jamming timeslot will be the blocking targets of the next jamming timeslot. The above malicious jamming has the characteristics of frequency tracking and selective jamming, which obviously has a certain degree of intelligence. In this article, we will call it intelligent multi-channel blocking jamming.
4) When multiple sensor nodes occupy the same channel for transmission, mutual interference will occur. Both mutual interference and intelligent multi-channel blocking jamming can cause transmission failure, while the effect of channel noise on transmission is ignored. Since all nodes are synchronized according to the timeslot, when a sensor node is sensing, other nodes are also performing the same operation, which makes it impossible to directly perceive mutual interference. However, if the node neither receives ACK message nor senses malicious jamming, it can be determined that the reason for the transmission failure is mutual interference.

B. PROBLEM FORMULATION

In traditional single-agent reinforcement learning, a Markov decision process (MDP) that includes a single agent and multiple environment states is generally used for problem formulating. However, in the multi-agent scenario considered in this article, the actions taken by any agent will have an impact on the state of the environment, as well as the rewards that can be obtained by other agents. This is a game involving multiple agents and multiple states. Therefore, extending MDP to multi-agent scenarios is a stochastic game, also known as Markov game [34], which can be used to model multi-agent reinforcement learning (MARL) problems. Mathematically, the anti-jamming problem can be expressed as a tuple $<N, \mathcal{S}, \mathcal{A}_1, \ldots, \mathcal{A}_N, f, R_1, \ldots, R_N>$, where the specific meanings of each element are as follows:

- N represents the number of sensor nodes;
- $\mathcal{S}$ represents the environment state space; $s \in \mathcal{S}$ is the element of the state space, representing the environment state of the WSN;

- $\mathcal{A}_n, n = 1, 2, \ldots, N$ represents the action space of sensor node n; $a_n \in \mathcal{A}_n$ is the optional action of sensor node n;
- $f : \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \times \mathcal{S} \to [0, 1]$ is the state transition probability function, which represents the probability that the environment state is transferred to s' after different nodes take action $a_n \in \mathcal{A}_n$ in state s;
- $R_a, n = 1, 2, \ldots, N$ represents the reward obtained after node n executes action $a_n \in \mathcal{A}_n$ in state s.

The environment state of the WSN is closely related to the jamming signal, and hence the environment state space is defined as follows:

$$\mathcal{S} \triangleq \{s : s = (j_1, \ldots, j_K)\}, \tag{1}$$

where $j_k \in \{1, \ldots, M\}, k = 1, \ldots, K$ represents the serial numbers of K blocked channels sensed by the sensor node through broadband spectrum. We represent the state s of the environment by arranging K different j_k from small to large. There are C_M^K states in the environment state space.

The action of each sensor node is to select its own transmission channel. Therefore, the independent action space of each sensor node is the same, i.e., $\mathcal{A}_1 = \mathcal{A}_2 = \cdots = \mathcal{A}_N$. Then, the independent action space of any node n can be defined as:

$$\mathcal{A}_n \triangleq \{a_n : a_n \in \{1, \ldots, M\}\}, \tag{2}$$

where independent action $a_n \in \{1, \ldots, M\}$ represents the number of the transmission channel selected by node n. A joint action $\mathbf{a} = \{a_1, \ldots, a_N\}$ is a combination of independent actions of different nodes, and hence the joint action space can be defined as follows:

$$\mathcal{A} \triangleq \mathcal{A}_1 \otimes \mathcal{A}_2 \otimes \cdots \otimes \mathcal{A}_N, \tag{3}$$

where $\otimes$ represents the cartesian product operation. There are C_{M+N-1}^N joint actions in the joint action space.

The transition of environment state depends on the change of jamming channel. As mentioned above, the change of jamming channel depends on the statistics and selection of the transmission channel by the intelligent jammer. Obviously, The transitions of environmental state are difficult to predict and model when the sensor nodes are not aware of the jamming strategy.

The local reward for node n taking independent actions a_n in state s depends on whether there are other nodes or jamming signals in the selected transmission channel, which can be defined as follows:

$$r_n(s, a_n) = \begin{cases} 1, & a_n \neq j_k \ \& \ a_n \neq a_m (m \in \mathcal{N}/n); \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The above formula means that when the data packet sent by node n is successfully received (confirmed by ACK message), the reward is 1, otherwise it is 0. Different nodes get the same reward by taking joint action $\mathbf{a} = \{a_1, \ldots, a_N\}$, which is the sum of local rewards of each node. It can be expressed as follows:

$$R_1(s, a) = \cdots = R_N(s, a) = R(s, a) = \sum\nolimits_{n=1}^{N} r_n(s, a_n). \tag{5}$$

In a stochastic game, agents may have cooperative, competitive or mixed relationships. Stochastic game can be divided into different categories according to different reward functions. When the reward function of all agents is exactly the same, there is a cooperative relationship between agents, which can be called a complete cooperative game. If the sum of reward functions of two agents is zero, there is a competitive relationship between them, which can be called a zero-sum game. When there are multiple types of reward functions among agents, there is a mixed relationship between agents, which can be called general and random games.

In the above stochastic game, different nodes clearly have a completely cooperative relationship, and their common goal is to obtain the optimal joint strategy Π^*. Each sensor node can obtain the largest cumulative discount reward for long-term execution of the optimal joint strategy Π^*. State-action value, also known as Q-value, can reflect the cumulative discount reward that a certain strategy can obtain [35], which can be defined as:

$$\mathcal{Q}^*(s, \mathbf{a}) = \max_{\pi} \mathbf{E}_{\pi} \left[\sum_{\tau=0}^{\infty} \gamma^{\tau} R_{t+\tau} | s_t = s, \ \mathbf{a}_t = \mathbf{a} \right], \tag{6}$$

where s_t and $\mathbf{a}_t$ are the state and joint action at step t, respectively. $R_{t+\tau}$ is the global immediate reward under strategy π at step $t + \tau$. $\mathbf{E}_{\pi}[\cdot]$ is the mathematical expectation operator. $0 \leq \gamma < 1$ is the discount factor which represents the importance of long-term reward [36].

If the optimal state-action values corresponding to all states-action pairs can be obtained, the optimal joint strategy can be introduced according to the optimal state-action value function as follows:

$$\Pi^*(\mathbf{a}|s) = \begin{cases} 1, & \text{if} \ \ \mathbf{a} = \arg\max_{\mathbf{a} \in \mathcal{A}} \mathcal{Q}^*(s, \mathbf{a}); \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

III. JOINT MULTI-AGENT ANTI-JAMMING ALGORITHM

A. DETAILED DESCRIPTION OF THE ALGORITHM

According to the analysis in the previous section, to obtain the optimal joint strategy, we need to obtain the optimal Q-values corresponding to all state-action combinations. Besides, since the state transition probability function is difficult to model, model-free reinforcement learning algorithm should be adopted to calculate the optimal Q-value. Q-learning algorithm is a classical model-free reinforcement learning algorithm, which can approach the optimal Q-values gradually through simple iteration [37]. Specifically, the Q-learning algorithm creates a Q-table to store the corresponding Q-values for all state-action pairs. In any given state, the algorithm selects an action according to the current Q-table. After performing the selected action, the algorithm observes the immediate reward and the next state, and then updates the Q-values based on the Q-value function. In the above-mentioned fully cooperative stochastic game, all nodes have the same reward function, and hence the state-action value function of each node executing any joint strategy

should also be the equal. In other words, all the nodes only need to update the same Q-table. The Q-values can be updated as follows:

$$\mathcal{Q}(s_t, \mathbf{a}_t) = \begin{cases} \mathcal{Q}(s_t, \mathbf{a}_t) + \alpha_t\big(R_t + \gamma \max_{\mathbf{a}_{t+1}} \mathcal{Q}(s_{t+1}, \mathbf{a}_{t+1}) \\ \quad - \mathcal{Q}(s_t, \mathbf{a}_t)\big), \quad \text{if} \quad s = s_t, \mathbf{a} = \mathbf{a}_t; \\ \mathcal{Q}(s_t, \mathbf{a}_t), \quad \text{otherwise.} \end{cases} \tag{8}$$

Besides, substituting Eq. (5) into Equation (6) can be obtained as follows:

$$\begin{aligned} \mathcal{Q}(s, \mathbf{a}) &= E\left[\sum_{\tau=0}^{\infty} \gamma^{\tau} R_{t+\tau} | s_t = s, \mathbf{a}_t = \mathbf{a}\right] \\ &= \sum_{n=1}^{N} E\left[\sum_{\tau=0}^{\infty} \gamma^{\tau} r_{n,t+\tau} | s_t = s, \mathbf{a}_t = \mathbf{a}\right] \\ &= \sum_{n=1}^{N} \mathcal{Q}_n(s, \mathbf{a}_n). \end{aligned} \tag{9}$$

In the above, $\mathcal{Q}_n(s, a_n)$ represents the Q-value corresponding to the independent action path (i.e., the independent strategy) of node n, which may be called independent Q-value. Then, Eq. (8) can be rewritten as:

$$\mathcal{Q}(s_t, \mathbf{a}_t) = \begin{cases} \sum_{n=1}^{N} \big(\mathcal{Q}_n\big(s_t, a_{n,t}\big) + \alpha_t\big(r_{n,t} \\ \quad + \gamma \max_{a_{t+1}} \mathcal{Q}_n\big(s_{t+1}, a_{n,t+1}\big) \\ \quad - \mathcal{Q}_n\big(s_t, a_{n,t}\big)\big)\big), \quad \text{if} \quad s = s_t, a_n = a_{n,t}; \\ \mathcal{Q}(s_t, \mathbf{a}_t), \quad \text{otherwise.} \end{cases} \tag{10}$$

Therefore, the update of the Q-value $\mathcal{Q}(s_t, \mathbf{a}_t)$ in joint Q-table can be converted into updating the independent Q-values $\mathcal{Q}_n(s_t, a_{n,t})$ of each sensor node separately as follows:

$$\begin{aligned} \mathcal{Q}_n\big(s_t, a_{n,t}\big) = \mathcal{Q}_n\big(s_t, a_{n,t}\big) + \alpha_t\big(r_{n,t} \\ + \gamma \max_{a_{t+1}} \mathcal{Q}_n\big(s_{t+1}, a_{n,t+1}\big) - \mathcal{Q}_n\big(s_t, a_{n,t}\big)\big), \end{aligned} \tag{11}$$

and then summing, thereby achieving distributed update of the Q-value $\mathcal{Q}(s_t, \mathbf{a}_t)$. When all Q-values in the joint Q-table converge to the optimal value, the nodes can obtain the optimal joint strategy according to Eq. (7).

According to the analysis above, we propose a joint multi-agent anti-jamming algorithm (JMAA) based on Q-learning. As illustrated in Fig. 2, each sensor node maintains an independent Q-table and the sink node maintains a joint Q-table. The rows and columns of the independent Q-table correspond to the environment states and independent actions, respectively. Therefore, the independent Q-table has C_M^K rows and M columns. Similarly, the rows and columns of the joint Q-table correspond to the environment states and joint actions, respectively. Therefore, the joint Q-table has C_M^K rows and C_{M+N-1}^N columns. The core idea of the proposed algorithm is that each node updates its independent Q-value according to local sensing results and transmission rewards, while the sink node accepts all independent Q-values to update the joint Q-value and decides the next transmission action of all nodes. In brief, the proposed JMAA has the characteristics of "distributed learning, centralized decision-making, and independent execution".

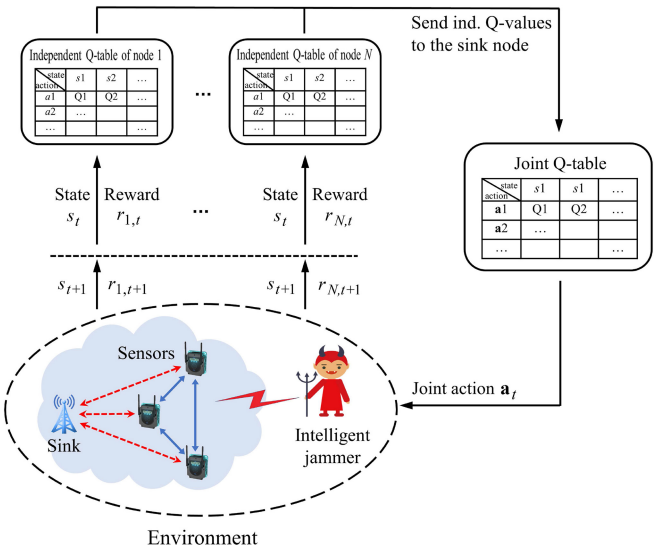


FIGURE 2. Multi-agent Q-learning based model.

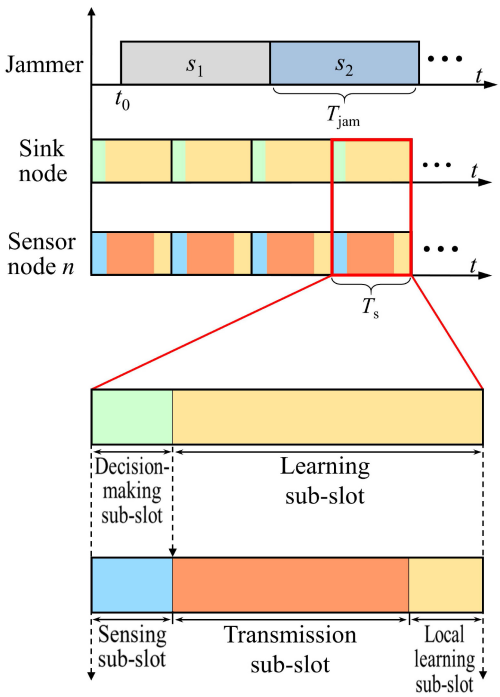


FIGURE 3. Illustration of the communication timeslot structure.

As shown in Fig. 3, the sink node divides the communication timeslot into decision-making sub-slot and learning sub-slot, which are used to coordinate the transmission channel and implement the learning algorithm respectively. The sensor node divides the communication timeslot into sensing sub-slot, transmission sub-slot and local learning sub-slot, which are respectively used for jamming sensing, data transmission and local learning. Each timeslot corresponds to an

Algorithm 1: Joint Multi-Agent Anti-Jamming Algorithm (JMAA)

1 **Initialize:** $\alpha, \gamma, \varepsilon \in [0, 1)$, $\mathcal{Q}(s_t, \mathbf{a}_t)$, $\mathcal{Q}_n(s_t, a_{n,t})$;
2 **for** $t = 1, \ldots, T$ **do**
3 Sensor nodes obtain state $s_t = (j_1, \ldots, j_K)$;
4 Sensor nodes transmit s_t and $\mathcal{Q}_n(s_{t-2}, a_{n,t-2})$ to the sink node;
5 The sink node selects a joint action $\mathbf{a}_t$ by the ϵ-greedy algorithm;
6 The sink node sends instructions to sensor nodes according to $\mathbf{a}_t$;
7 Sensor nodes perform independent action $a_{n,t}$ according to the instructions;
8 Sensor nodes calculate reward $r_n(s_t, a_{n,t})$;
9 Sensor nodes update the independent Q-value $\mathcal{Q}_n(s_{t-1}, a_{n,t-1})$ by Eq. (11);
10 The sink node updates the joint Q-value $\mathcal{Q}(s_{t-1}, \mathbf{a}_{n,t-1})$ by Eq. (9);
11 $t = t + 1$.

iteration of the JMAA. The details of the JMAA is provided in Algorithm 1. The specific flow of the JMAA is as follows.

1) Firstly, in the sensing sub-slot, each sensor node obtains the current environment state s_t by jamming sensing (line 3), and then transmits s_t and the locally updated independent Q-value of the previous timeslot to the sink node together (line 4).
2) Secondly, in the decision-making sub-slot, the sink node selects a joint action by Softmax algorithm based on the current joint Q-table (line 5), and then sends instructions to all the sensor nodes to coordinate their transmission channels (line 6).
3) Thirdly, in the transmission sub-slot, each sensor node executes its own independent actions according to the instructions from the sink node, i.e., data transmission is carried out in the assigned channel respectively (line 7).
4) Lastly, in the local learning sub-slot, Each sensor node calculates its own reward based on the sensing results and the ACK message (line 8), then updates the independent Q-value by Eq. (11) (line 9).
5) While the sensor node performs the above two steps, in the learning sub-slot, the sink node updates the joint Q-value by Eq. (9) based on the independent Q-values updated in the previous timeslot (line 9).

In step 2 above, the Softmax algorithm is introduced to select the joint action, which is one of the common method to solve the "Exploration-Exploitation dilemma [38]" faced by reinforcement learning. Specifically, the strategy of selecting the joint action of the sink node can be expressed as:

$$\Pi(\mathbf{a}_t|s_t) = \frac{e^{\mathcal{Q}(s_t,\mathbf{a}_t)/\xi}}{\sum_{\mathbf{a}} e^{\mathcal{Q}(s_t,\mathbf{a})/\xi}} \tag{12}$$

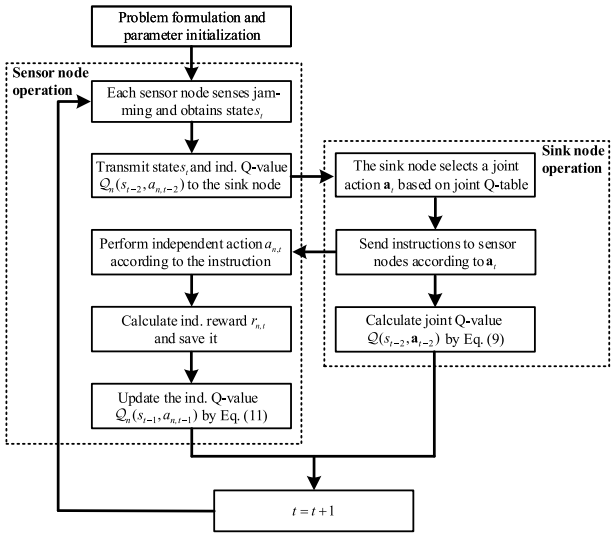


FIGURE 4. Flowchart summarizing the operations of nodes.

where $\xi > 0$ is called "temperature". The smaller ξ is, the greater the probability that the joint action with higher Q-value will be selected. As ξ approaches 0, the Softmax algorithm will tend to "exploit only". Conversely, as ξ tends to infinity, the Softmax algorithm tends to "explore only." To achieve a smooth transition from "exploration" to "exploitation", the temperature is updated according to the following rules:

$$\xi = \begin{cases} \xi_0 e^{-\upsilon t} & \xi \geq \xi_{final} \\ \xi_{final} & \xi < \xi_{final} \end{cases} \tag{13}$$

where the initial temperature ξ_0 is positively correlated with the "exploration" ability of the algorithm at the initial stage. When $\upsilon > 0$, ξ can approach 0 gradually with the algorithm iteration, and its value determines the length of the transition time.

Operations of nodes presented in the form of a flowchart in Fig. 4. Different from the offline algorithm, which needs to complete the training before output the strategy, the proposed JMAA is online, and its iterative learning process is also a process of constantly improving the transmission strategy. It means that sensor nodes and the sink node will continue to execute the proposed algorithm until the transmission is terminated. As the transmission progresses, the joint Q-table is continuously updated, i.e., the transmission strategy is continuously improved. After a finite number of iterations, when all Q-values in the joint Q-table do not change significantly, it means that the Q-values have converged to the optimum. The strategy based on the joint Q-table converges to the optimal strategy.

B. COMPLEXITY ANALYSIS

The main computational complexity of the proposed Algorithm 1 lies in steps 3 to 7. The steps 3 to 7 are performed only once in each iteration, and their computational complexity is independent of the size of the Q-table.

Hence, the computational complexity of step 5, 6 and 10 of the sink node can be expressed as $\mathcal{O}(3T)$. The computational complexity of each sensor node can be expressed as $\mathcal{O}(5T)$, then the computational complexity of N sensor nodes is $N \cdot \mathcal{O}(5T)$. The total computational complexity of Algorithm 1 can be expressed as $C = (N+1) \cdot \mathcal{O}(T)$, which means that the proposed algorithm can achieve an optimal solution in polynomial time.

As previously mentioned, the size of independent Q-table and joint Q-table are $C_M^K \times M$ and $C_M^K \times C_{M+N-1}^N$, respectively. Therefore, the space complexity of the sensor node and the sink node can be expressed as $\mathcal{O}(C_M^K \times M)$ and $\mathcal{O}(C_M^K \times C_{M+N-1}^N)$, respectively. The total space complexity of Algorithm 1 can be expressed as $\mathcal{O}(C_M^K \times M + C_M^K \times C_{M+N-1}^N)$, which means that the space complexity of Algorithm 1 will increase sharply with the number of channels and sensor nodes.

C. CONVERGENCE ANALYSIS

The authors in [36] have proved that when the learning rate α_t in Eq. (10) and (11) satisfies the following conditions:

$$\alpha_t \in [0, 1), \sum_{t=1}^{\infty} \alpha_t = \infty, \text{ and } \sum_{t=1}^{\infty} (\alpha_t)^2 < \infty, \quad (14)$$

Q-learning algorithm can traverse all states with the number of iterations increases, and finally converge to the optimal Q-values for all state-action pairs after a finite number of iterations. The proposed JMAA obtains the joint actions according to the joint Q-table, and hence it can converge to the optimal strategy.

D. SIGNALING OVERHEAD ANALYSIS

Since the proposed JMAA relies on the information interaction between the sink node and sensor nodes, the signaling overhead should be considered. As previously mentioned, in each iteration, the sensor node sends sensing result and independent Q-value to the sink node, and receives channel assignment instructions from the sink node. Let I_s, I_q and I_a denote the quantity of information contained in sensing result, in independent Q-value and in channel assignment instruction, respectively. The signaling overhead of each sensor node can be expressed as $(I_s + I_q)/T_s$, while the signaling overhead of the sink node can be expressed as I_a/T_s. Since N sensor nodes have to send information to the sink node in each iteration, the signaling overhead of N sensor nodes can be expressed as $[N(I_s + I_q)]/T_s$. Therefore, the total signaling overhead of Algorithm 1 can be expressed as $[N(I_s + I_q) + I_a]/T_s$, which means that the total signaling overhead is proportional to the number of sensor nodes.

IV. SIMULATION RESULTS

A. SIMULATION SETTING

The simulation parameter settings are shown in Table 1.

To evaluate the performance of the proposed JMAA, we compare the performance of the proposed algorithm with the following methods:

TABLE 1. Parameter settings.

Quantity	Symbol	Value
Number of Sensor nodes	N	3
Number of Channels	M	10
Jammer starting operating time	t_0	0.24ms
Jamming timeslot length	t_{jam}	1.5ms
Number of jammer blocked channels	K	3
Communication timeslot length	t_s	0.3ms
Sensing sub-slot length	t_1	0.03ms
Transmission sub-slot length	t_2	0.2ms
Local learning sub-slot length	t_3	0.07ms
Decision-making sub-slot length	t_4	0.03ms
Learning sub-slot length	t_5	0.27ms
Learning rate	α_t	0.8
Discount factor	γ	0.6
Initial temperature	ξ_0	100
Threshold temperature	ξ_{final}	0.02

- Frequency-hopping based method: The sensor nodes switch transmission channels according to the randomly generated fixed frequency-hopping patterns, and the frequency-hopping patterns of different sensor nodes are orthogonal to each other to ensure that the same channel is occupied by only one sensor node at the same time.
- Sensing based method: Each sensor node can sense all the jammed channels. If the channel in use is blocked in the current timeslot, the sensor node will randomly switch to an idle channel in the next timeslot, otherwise leaving the channel unchanged. Furthermore, there is no exchange of information among nodes.
- Independent Q-learning method (IQL): Each sensor node performs a Q-learning algorithm individually. Moreover, the decisions of each sensor are based solely on locally learning results, and the ACK mechanism is not adopted.
- Independent Q-learning method with ACK mechanism (IQL-ACK): This method introduces the ACK mechanism on the basis of IQL, and can determine whether there is mutual interference by combining with the result of jamming sensing. The difference between IQL-ACK and JMAA is that there is no information exchange among nodes in IQL-ACK, and each sensor node's decision is based on the local independent Q-table rather than the joint Q-table.
- Distributed Q-learning method (DQL): Each node adopts a multi-agent reinforcement learning algorithm called distributed Q-learning [39]. Similar to IQL, this method does not require information exchange between sensors. Each node maintains local Q-value $\mathcal{Q}_n(s_t, a_{n,t})$ through its own actions and rewards. The update of Q-value is carried out in the direction of increasing Q-value:

$$\mathcal{Q}_n(s_t, a_{n,t}) = \max\Big\{\mathcal{Q}_n(s_t, a_{n,t}), r_{n,t} + \gamma \max_{a_{t+1}} \mathcal{Q}_n(s_{t+1}, a_{n,t+1})\Big\}, \quad (15)$$

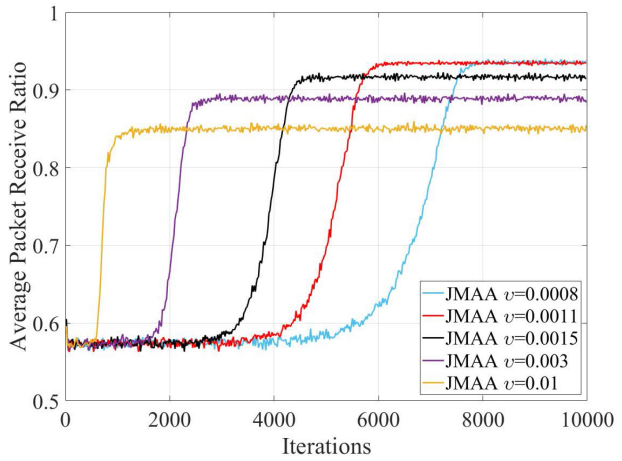


FIGURE 5. Average packet receive ratio of JMAA under different parameter settings.

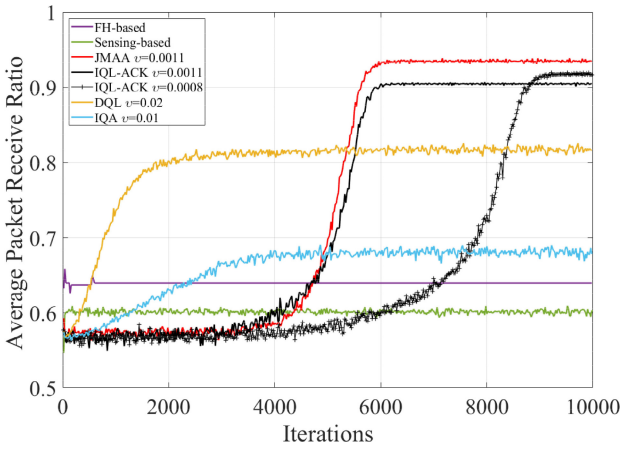


FIGURE 6. Average packet receive ratio of JMAA under different parameter Settings.

We introduce the average packet receive ratio to compare the anti-jamming performance of different methods. The average packet receive ratio can be defined as $\rho_{avg}(t) = 1/N \sum_{n=1}^{N}(D_n(t)/W)$, W is the number of independent runs of the proposed algorithm. $D_n(t)$ is the number of data packets successfully transmitted by sensor node n in timeslot t when the algorithm runs independently W times. Besides, the following simulation results about the average packet receive ratio are the average of 5000 independent runs.

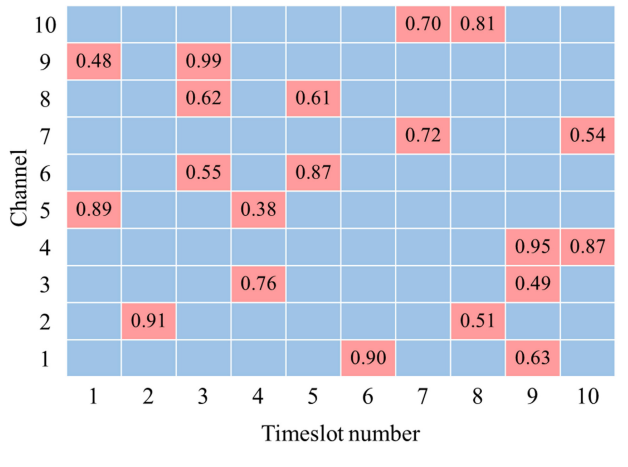
B. SIMULATION ANALYSIS

Fig. 5 compares the average packet receive ratio of JMAA when Softmax algorithm has different parameters. The smaller υ is, the longer the exploration process of JMAA is, which means that the convergence rate is slower. However, sufficient exploration can make the convergence value of the average packet receive ratio higher. When it takes at least 6000 iterations to complete the transition from exploration to exploitation, the average packet receive ratio of JMAA can converge to the optimal value, about 0.93.

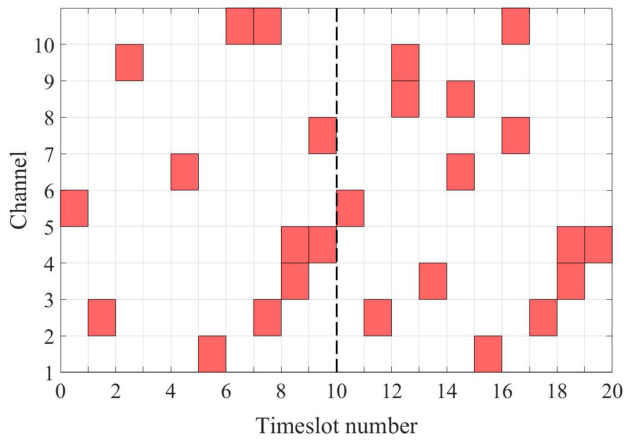
Fig. 6 shows a comparison of the average packet receive ratio for different methods. Considering that the performance of anti-jamming algorithm based on reinforcement learning is affected by parameter setting of Softmax algorithm, we choose the optimal performance curve of DQL and IQL (i.e., the case with the least number of iterations when the optimal performance is achieved) as the comparison scheme. Besides, due to the excessive number of iterations required for IQL-ACK to converge to the optimal value, the performance curve when the convergence is completed within 10,000 iterations is shown in Fig. 6. Firstly, it is known from Fig. 5 that the average packet receive ratio of JMAA can converge to the optimal value of 0.93 within 6000 iterations, while the IQL-ACK can converge to 0.9 within 6000 iterations and 0.915 within 9000 iterations. It means that IQL-ACK needs more iterations to achieve similar performance to JMAA. The reason is that cooperative learning among nodes is not introduced in IQL-ACK, and it takes more time for nodes to independently explore and find the optimal strategy. Secondly, although the optimal performance curves of DQL and IQL converge quickly, the optimal value after convergence is significantly worse than that of JMAA. The reason is that updating Q-value of DQL according to Eq. (15) can always make it proceed in the direction of increase, which has the advantage of accelerating convergence and the disadvantage of falling into local optimization. IQL ignores the mutual interference among nodes, resulting in fast convergence but poor anti-jamming effect. Finally, due to the fixed anti-jamming strategy, the average packet receive ratio of FH-based and Sense-based methods commonly used in practice is far lower than that of JMAA, and even lower than the above several comparison algorithms based on reinforcement learning.

Since the proposed JMAA does not need to model the jamming patterns, and has the ability to explore and learn from the unknown jamming environment, it should be able to solve the problem of reliable communication in various jamming environments. Hence, the following simulation verify the performance of the proposed JMAA when the external malicious jamming is sweep jamming or probabilistic jamming [40]. Among them, the sweep jamming is a conventional dynamic jamming which can periodically jam the target frequency range or the target channel in turn. Moreover, probabilistic jamming can determine the target channel of different timeslots according to the specific jamming probability matrix. To be specific, if the jammer determines the jamming channels according to the probability matrix shown in Fig. 7(a), then Fig. 7(b) shows the generated jamming pattern in two jamming cycles. More details about probabilistic jamming can be found in [39].

Fig. 8 and Fig. 9 show the average packet receive ratio of JMAA in probabilistic jamming environment and sweep jamming environment. In both sweeping and probabilistic jamming environment, with the decrease of the parameter υ, the convergence speed decreases, but the convergence value is closer to 1. Obviously, the average packet receive ratio can



(a)



(b)

FIGURE 7. Diagram of probabilistic jamming: (a) Jamming probability matrix. (b) Generated jamming pattern.

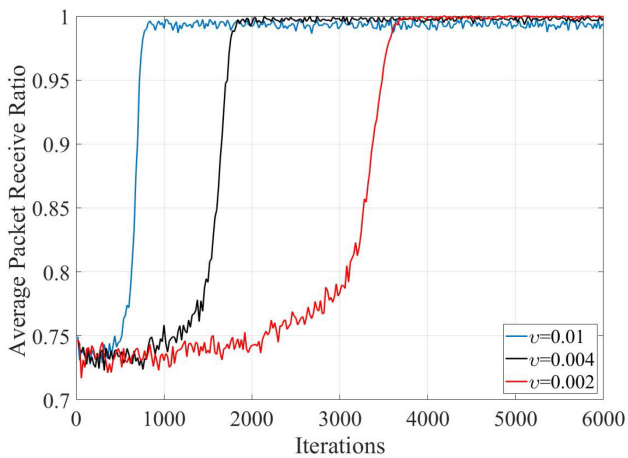


FIGURE 8. Average packet receive ratio of JMAA in probabilistic jamming environment.

converge to 1 when the appropriate parameter is set, which means that the proposed JMAA can completely avoid the malicious jamming and mutual interference. In addition, the average packet receive ratio of JMAA requires at least 4000 iterations of exploration before it converges to 1 in probabilistic jamming environment, while in the sweep jamming environment, it only needs 1000 iterations of exploration.

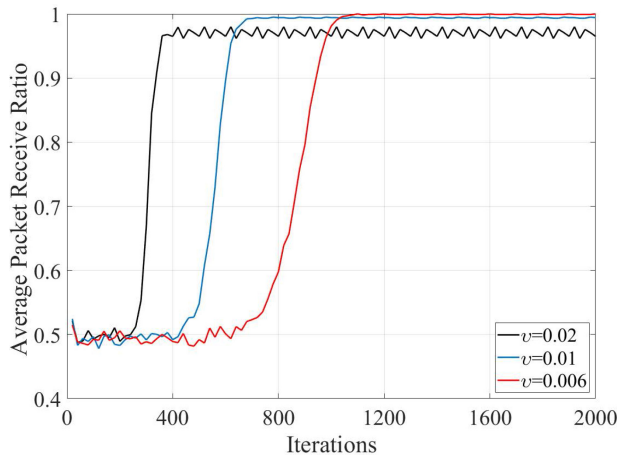


FIGURE 9. Average packet receive ratio of JMAA in sweep jamming environment.

V. CONCLUSION

In this article, we investigate the problem of anti-jamming communication in a wireless sensor network. For the internal mutual interference caused by competition among sensor nodes and external intelligent multi-channel blocking jamming. We model the anti-jamming problem as a stochastic game framework, and a joint multi-agent anti-jamming algorithm (JMAA) is proposed for achieving real-time anti-jamming channel selection. By cooperative learning, the proposed JMAA can eliminate mutual interference and effectively avoid the tracking of intelligent multi-channel blocking jamming. The simulation results show that the proposed JMAA is superior to the frequency-hopping based method, the sensing-based method and the independent Q-learning method (with or without ACK mechanism). In addition, we prove the effectiveness of the proposed JMAA in sweep jamming or probabilistic jamming environment, which indicates the proposed JMAA can be widely used in various of jamming environments.

In future work, the transfer learning approach may be a good candidate to obtain faster convergence speed in multi-user sensor networks with limited computing resources. In addition, it would be more meaningful to consider that different nodes face different external jamming.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, 2008.

[3] D. Bruckner, C. Picus, R. Velik, W. Herzner, and G. Zucker, "Hierarchical semantic processing architecture for smart sensors in surveillance networks," *IEEE Trans. Ind. Informat.*, vol. 8, no. 2, pp. 291–301, May 2012.

[4] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. Hanzo, "Network-lifetime maximization of wireless sensor networks," *IEEE Access*, vol. 3, pp. 2191–2226, 2015.

[5] G. Han, J. Jiang, N. Bao, L. Wan, and M. Guizani, "Routing protocols for underwater wireless sensor networks," *IEEE Commun. Mag.*, vol. 53, no. 11, pp. 72–78, Nov. 2015.
[6] B. Rashid and M. H. Rehmani, "Applications of wireless sensor networks for urban areas: A survey," *J. Netw. Comput. Appl.*, vol. 60, pp. 192–219, Jan. 2016.
[7] A. Milenkovi, C. Otto, and E. Jovanov, "Wireless sensor networks for personal health monitoring: Issues and an implementation," *Comput. Commun.*, vol. 29, nos. 13–14, pp. 2521–2533, 2006.
[8] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 266–282, 1st Quart., 2014.
[9] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 2, pp. 245–257, 2nd Quart., 2011.
[10] M. Young and R. Boutaba, "Overcoming adversaries in sensor networks: A survey of theoretical models and algorithmic approaches for tolerating malicious interference," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 617–641, 4th Quart., 2011.
[11] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proc. IEEE*, vol. 75, no. 1, pp. 56–73, Jan. 1987.
[12] G. Heidari-Bateni and C. D. Mcgillem, "A chaotic direct-sequence spread-spectrum communication system," *IEEE Trans. Commun.*, vol. 42, no. 2, pp. 1524–1527, Feb.–Apr. 1994.
[13] H. Zhu, C. Fang, Y. Liu, C. Chen, M. Li, and X. S. Shen, "You can jam but you cannot hide: Defending against jamming attacks for geo-location database driven spectrum sharing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2723–2737, Oct. 2016.
[14] L. Zhang, Z. Guan, and T. Melodia, "United against the enemy: Anti-jamming based on cross-layer cooperation in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 8, pp. 5733–5747, Aug. 2016.
[15] Q. Peng, P. C. Cosman, and L. B. Milstein, "Spoofing or jamming: Performance analysis of a tactical cognitive radio adversary," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 903–911, Apr. 2011.
[16] M. Balakrishnan, H. Huang, R. Asorey-Cacheda, S. Misra, S. Pawar, and Y. Jaradat, "Measures and countermeasures for null frequency jamming of on-demand routing protocols in wireless ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 11, pp. 3860–3868, Nov. 2012.
[17] M. Acharya and D. Thuente, "Intelligent jamming attacks counterattacks and (counter) attacks in 802.11b wireless networks," in *Proc. OPNETWORK Conf.*, 2005, pp. 1–12.
[18] F. Tang, Z. M. Fadlullah, B. Mao, and N. Kato, "An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: A deep learning approach," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5141–5154, Dec. 2018.
[19] F. Tang, B. Mao, Z. M. Fadlullah, J. Liu, and N. Kato, "ST-DeLTA: A novel spatial-temporal value network aided deep learning based intelligent network traffic control system," *IEEE Trans. Sustain. Comput.*, vol. 5, no. 4, pp. 568–580, Oct./Dec. 2020.
[20] J. Nie and S. Haykin, "A Q-learning-based dynamic channel assignment technique for mobile communication systems," *IEEE Trans. Veh. Technol.*, vol. 48, no. 5, pp. 1676–1687, Sep. 1999.
[21] M. Bennis and D. Niyato, "A Q-learning based approach to interference avoidance in self-organized femtocell networks," in *Proc. IEEE Globecom Workshops*, Miami, FL, USA, 2010, pp. 706–710.
[22] F. Tang, Y. Zhou, and N. Kato, "Deep reinforcement learning for dynamic uplink/downlink resource allocation in high mobility 5G HetNet," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2773–2782, Dec. 2020.
[23] N. Abuzainab *et al.*, "QoS and jamming-aware wireless networking using deep reinforcement learning," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Norfolk, VA, USA, 2019, pp. 610–615.
[24] T. Erpek, Y. E. Sagduyu, and Y. Shi, "Deep learning for launching and mitigating wireless jamming attacks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 1, pp. 2–14, Mar. 2019.
[25] C. Zhong, F. Wang, M. C. Gursoy, and S. Velipasalar, "Adversarial jamming attacks on deep reinforcement learning based dynamic multichannel access," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Seoul, South Korea, 2020, pp. 1–6.
[26] B. Wang, Y. Wu, K. J. R. Liu, and T. C. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 877–889, Apr. 2011.
[27] S. Machuzak and S. K. Jayaweera, "Reinforcement learning based anti-jamming with wideband autonomous cognitive radios," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, 2016, pp. 1–6.
[28] F. Slimeni, Z. Chtourou, B. Scheers, V. L. Nir, and R. Attia, "Cooperative Q-learning based channel selection for cognitive radio networks," *Wireless Netw.*, vol. 25, no. 4, pp. 4161–4171, 2018.
[29] S. Singh and A. Trivedi, "Anti-jamming in cognitive radio networks using reinforcement learning algorithms," in *Proc. 9th Int. Conf. Wireless Opt. Commun. Netw. (WOCN)*, 2012, pp. 1–5.
[30] F. Slimeni, B. Scheers, Z. Chtourou, and V. L. Nir, "Jamming mitigation in cognitive radio networks using a modified Q-learning algorithm," in *Proc. Int. Conf. Mil. Commun. Inf. Syst. (ICMCIS)*, 2015, pp. 1–7.
[31] M. A. Aref, S. K. Jayaweera, and S. Machuzak, "Multi-agent reinforcement learning based cognitive anti-jamming," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2017, pp. 1–6.
[32] F. Yao and L. Jia, "A collaborative multi-agent reinforcement learning anti-jamming algorithm in wireless networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1024–1027, Aug. 2019.
[33] X. Liu, Y. Xu, L. Jia, Q. Wu, and A. Anpalagan, "Anti-jamming communications using spectrum waterfall: A deep reinforcement learning approach," *IEEE Commun. Lett.*, vol. 22, no. 5, pp. 998–1001, May 2018.
[34] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 2, pp. 156–172, Mar. 2008.
[35] Z. Zhou, *Machine Learning (in Chinese)*, Tsinghua Univ. Press, Beijing, China, 2016.
[36] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.
[37] N. V. Huynh, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, "'Jam me if you can': Defeating jammer with deep dueling neural network architecture and ambient backscattering augmented communications," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2603–2620, Nov. 2019.
[38] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 237–285, 1996.
[39] M. Lauer and M. A. Riedmiller, "An algorithm for distributed reinforcement learning in cooperative multi-agent systems," in *Proc. 17th Int. Conf. Mach. Learn.*, San Francisco, CA, USA, 2000, pp. 535–542.
[40] Y. Wang, Y. Niu, J. Chen, F. Fang, and C. Han, "Q-learning based adaptive frequency hopping strategy under probabilistic jamming," in *Proc. 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, 2019, pp. 1–7.

QUAN ZHOU was born in Liyang, China, in 1991. He received the B.S. degree in communication engineering from East China Jiaotong University, Nanchang, in 2014. He is currently pursuing the master's degree in electronics and communication engineering with the University of Army Engineering University of PLA, with a focus in intelligent communication anti-jamming method.

YONGGUI LI was born in Anhui, China, in 1964. He received the M.S. degree in information and communication engineering from the PLA University of Science and Technology in 2000.

He is currently a Senior Research Fellow with the National University of Defense Technology, China. He has authored or coauthored more than 80 journal and conference papers, and published one book. He is currently engaged in the research of modern wireless communication and its network intelligence theory and technology, especially wireless communication spectrum sensing and jamming analysis, dynamic spectrum access, and adaptive communication algorithm.

YINGTAO NIU was born in Taian, China, in 1978. He received the M.S. degree from PLA Commanding Communication Academy, China, in 2005, and the Ph.D. degree from the Institute of Communication Engineering, PLA University of Science and Technology, China, in 2008.

He is currently a Senior Research Fellow with the National University of Defense Technology, China. He has authored more than 40 journal and conference papers. His main research interests are cognitive radio theory and techniques, with particular emphasis on algorithms of signal sensing and communication decision-making algorithm in cognitive radio systems.