IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 October 2022, accepted 12 October 2022, date of publication 14 October 2022, date of current version 28 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3214842

RESEARCH ARTICLE

Multifunctional Radar Cognitive Jamming Decision Based on Dueling Double Deep Q-Network

LU-WEI FENG, SONG-TAO LIU, AND HUA-ZHI XU
Department of Information System, Dalian Naval Academy, Dalian 116018, China
Corresponding author: Song-Tao Liu (navylst@163.com)

This work was supported in part by the China Postdoctoral Science Foundation under Grant 2015M572694 and Grant 2016T90979.

ABSTRACT To solve the inefficient and imprecise problem using the Deep Q-network (DQN) algorithm for the radar jamming decision, this paper proposes a multifunctional radar jamming decision optimization method based on the Dueling Double Deep Q-network (D3QN). First, we use a value function reflecting the radar state's change, and an advantage function related to radar state S and jamming action A to improve the cognitive jamming level for unknown radar modes. Then, using the dueling networks for jamming strategy selection and effectiveness evaluation can further improve decision accuracy. Finally, we propose a prioritized experience replay mechanism during network training to shorten the decision-making time. The experimental results show that our proposed method completes decision tasks 2.1 times more efficiently than the DQN and improves decision accuracy by approximately 10% over DQN.

INDEX TERMS Dueling double deep q-network (D3QN), prioritized experience replay, jamming decision-making, reinforcement learning.

I. INTRODUCTION

As multifunctional radars with complex parameter systems continue to be put into the modern battlefield constantly, the struggle between the radar and jamming sides is undergoing an unprecedented change. The radar detection technology is always ahead of the development of electronic jamming technology [1], [2], which shortens the jamming sides' response time. This situation makes it difficult for the current jamming decision technology to countermeasure the emerging modern radar, such as multifunctional radars and cognitive radar [3], [4]. Therefore, it is urgent to study the improved method for the radar jamming decision.

In recent years, the rapid development of artificial general intelligence (AGI) technology has given rise to many advanced techniques and optimization theories [5]. Reinforcement learning, an important branch of machine learning, is considered one of the essential directions of AGI research [6]. DeepMind proposed Deep Q-Network (DQN) in 2013, combining neural networks and Q-learning with building an end-to-end control policy model and successfully validating the method's feasibility in Atari games [7]. Nature DQN algorithm was proposed again in 2015, establishing its leadership in the field of reinforcement learning with excellent empirical results [8]. Currently, it has been widely used in game competitions [9], [10], decision optimization [11], scheduling control [12], [13], and many other areas.

Due to the powerful function of reinforcement learning methods, many scholars have proposed various radar jamming decision methods based on reinforcement learning theory. They offer great potential for promoting autonomy and intelligence in the radar countermeasure process [14], [15]. Li et al. [16] introduced cognitive techniques into the radar countermeasure process for the first time, providing a new idea for radar jamming decisions. Xing et al. [17], [18] further analyzed the Q-learning theory and solved the problems of jamming decisions when the radar operating mode is unknown. Li et al. [19] improved the Q-learning theory with the Simulated Annealing (SA) algorithm to enhance jamming strategy exploration and utilization. Zhang et al. [20] used the reinforcement learning method to make the jamming decision process more scientific and rational. Gao et al. [21] established an offensive and defensive model for jamming against the cognitive radar, the

The associate editor coordinating the review of this manuscript and approving it for publication was Francisco J. Garcia-Penalvo.

dynamic process is realized to find a reasonable jamming strategy. Smits et al. [22] presented a cognitive radar network that uses available resources, sharing the data among the network components and considering prior knowledge for jamming decisions. Pan et al. [23] applied the improved chaotic genetic algorithm to allocate jamming strategies and evaluated the jamming effect with the radar detection probability as the index. Liu et al. [24] solved the jamming strategy allocation problem by comparing the differences between the Q-learning algorithm and Double Deep Q-Network (DDQN) algorithm. The above radar jamming decision methods have achieved the desired results to a certain extent. However, there are still problems of slow decision speed and low accuracy due to the increased number of radar modes [18], [19], [20], [24].

This paper proposes a multifunctional radar jamming decision method based on Dueling Double Deep Q-Network (D3QN) to solve the above problems. We first analyze the shortcomings of the traditional method in solving the jamming decision problem, and then establish a D3QN-based decision model according to the operational characteristics of multifunctional radar. Next, we use the DDQN to solve the problem of Q-value overestimation. Then, we adopt the dueling networks to calculate the Q values for jamming actions more accurately, reducing the error of values in complex countermeasures environments. Finally, we propose a prioritized experience replay mechanism to improve the sample utilization and reduce the decision-making time further. The simulation results show that the D3QN method has apparent advantages in decision efficiency and jamming accuracy.

The paper is arranged as follows. Section 2 analyzes the inefficient overestimation problem of traditional reinforcement learning methods and introduces our method. In Section 3, we describe the core technology of the D3QN method. The simulation results are shown in Section 4, in which the scientific and feasibility of the proposed method are demonstrated. Finally, we provide a conclusion in Section 5.

II. RADAR JAMMING DECISION METHOD BASED ON REINFORCEMENT LEARNING

A. REINFORCEMENT LEARNING PRINCIPLES FOR JAMMING DECISION-MAKING

Reinforcement learning uses the ''trial and error'' mechanism in psychology. The agent obtains an evaluative reward signal through continuous interaction with the unknown environment and repeats this process to generate optimal strategies [25]. Even if the agent does not have prior knowledge of the environment, it can still learn the best strategy through the decision-making process, which becomes one of the effective methods to solve the decision-making problem of nonlinear stochastic systems [26].

The specific process of radar jamming decisions using reinforcement learning method is as follows: the jammer detects the target radar and obtains the information of the

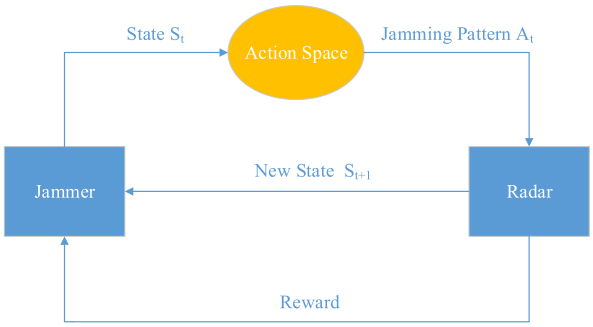

FIGURE 1. Radar jamming decision process based on reinforcement learning.

radar state s_t, and $s_t \in S$ denotes the set of states of the radar at the t moment, S represents the set of all operating states of the radar. The jammer is given a feedback reward by performing a jamming action on the target radar, $a_t \in A$ denotes the set of jamming actions that the jammer can take. At this point the radar is shifted to a new state s_{t+1} due to jamming. In the process of continuous countermeasure, the mapping function from the radar state to the jamming action is defined as a strategy $\boldsymbol{\pi} : S \rightarrow A$. The jammer can calculate the value of a strategy based on the feedback reward and use it as the basis for selecting the optimal strategy. As shown in Fig. 1. By repeating the above process, the value function $V_\pi(s_t)$ of the strategy π, which means the sum of the feedback rewards from the t moment, can be obtained

$$\begin{aligned} V_\pi(s_t) &= E\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots\right] \\ &= E\left[\sum_{i=0}^{\infty} \gamma^i (r_{t+i})\right] \end{aligned} \tag{1}$$

where $\gamma \in [0, 1]$, denotes the reward discount rate of the learning process. Thus for all the strategies π have a value function corresponding to them, the value function of the optimal strategy π^* can be found

$$V_*(s_t) = \max V_\pi(s_t) \tag{2}$$

B. DQN METHOD

Q-learning is a derivative reinforcement learning theory, enabling decision optimization by establishing a dynamic programming process.When the problem is characterized by a Markov process, the future state is only related to the current state and not to the past state. According to the Bellman equation, the state action-value function of traditional Q-learning can be expressed as

$$\begin{aligned} Q(s_t, a_t) = R(s_t, a_t) + \alpha [r_{t+1} + \gamma \max Q(s_{t+1}, a_t) \\ - Q(s_t, a_t)] \end{aligned} \tag{3}$$

where $Q(s_t, a_t)$ indicates that when the target radar is in the state s_t, the sum of rewards obtained by the jammer after taking a jamming action a_t, $\alpha \in [0, 1]$ is the learning rate. The optimal decision is output when the expected sum converges, which is suitable for decision problems with simple space and

low dimensionality. When the number of target radar states increases, the high-dimensional radar states make the size of the relationship Q-table tremendous. The complexity of the algorithm grows exponentially to the ''the curse of dimensionality'' problem [27], leading to a significant decrease in the overall decision efficiency. As a result, it is difficult to apply to the multifunctional radar jamming decision problems effectively.

DQN differs from the traditional Q-learning by fitting Q-learning with the deep neural networks. It outputs Q-values directly from the high-dimensional raw data using two neural networks with the same structure and different parameters [28], [29]. The value network reflects the real reward value obtained by the jammer interacting with the target radar, denoted as

$$Q(s, a; \theta) = E_{(s,a)}\left[r + \gamma \max Q\left(s', a'; \theta'\right) | s, a\right] \quad (4)$$

The estimation network uses the sample data to estimate the state action-value $Q\left(s, a; \theta'\right)$, introducing a loss function $L(\theta)$ that represents the difference between the estimated value and the real value

$$L(\theta) = E_{(s,a,r,s')}\left[Q(s, a; \theta) - Q\left(s, a; \theta'\right)^2\right] \quad (5)$$

By training the sample and going through several iterations, the parameters θ' of the estimation network are continuously assigned to the value network. As a result, the real value is infinitely close to the estimated value so that the loss function is minimized, which makes the network more stable and solves the problem of ''the curse of dimensionality''. It opens up new research ideas and methods for the jamming decision-making problem of multifunctional radar.

C. D3QN METHOD

At present, DQN application in radar jamming decisions has achieved remarkable results [18], [19], [20], [24]. Nevertheless, analyzing the network structure and algorithm principle, the following aspects still deserve to be explored in depth.

(1) The model uses the same structure to generate the real reward and estimated values. However, when the network parameters are constantly updated, obtaining relatively stable estimated values is difficult, which is adverse to the algorithm's convergence.
(2) There is an estimation bias in the value function during training. Using $\max Q\left(s', a'; \theta'\right)$ can lead the model to overestimate the reward of action, thus misleading the jammer to choose the wrong action and fall into the locally optimal solution.

In order to solve the problem of DQN training instability and overestimation, this paper proposes the D3QN theory to improve the efficiency and accuracy of jamming decision-making. We first introduces the DDQN [30] based on DQN. DDQN represents the action selection and effectiveness evaluation as an estimation value network $Q_{\mathrm{M}}(s, a; \theta)$ and a target value network $Q_{\mathrm{T}}\left(s, a; \theta'\right)$. The estimation value network calculates the Q-value after jamming, the parameters θ are updated according to the sample. The target value network calculates the target value Y through time-series differential, the parameters θ' are replaced with the latest θ. Finally, the target value$Y$ is calculated as

$$Y = E_{(s,a,r,s')}\left[r + \gamma Q_{\mathrm{T}}\left(s', \operatorname{argmax} Q_{\mathrm{M}}\left(s', a'; \theta\right); \theta'\right)\right] \quad (6)$$

θ' holding constant for a period can make the target value Y relatively fixed, which is beneficial for convergence. We use the Q_{M} to generate the actions and the Q_{T} to calculate the target value. The maximum functions are not the same, preventing the model from selecting the sub-optimal actions that are overestimated. It effectively solves the overestimation problem of the DQN method.

D3QN takes advantage of the Dueling network architecture by diverting the estimation value network $Q_{\mathrm{M}}(s, a; \theta)$ of DDQN into two parts. The state value function $V\left(s; \theta, w^V\right)$ characterizes the influence of the radar state. The action advantage function $A\left(s, a; \theta, w^A\right)$ distinguishes the jamming effect in a given radar state [31]. The improved estimation value network $Q_{\mathrm{M}}\left(s, a; \theta, w^V, w^A\right)$ is defined as the following equation

$$Q_M\left(s, a; \theta, w^V, w^A\right) = V\left(s; \theta, w^V\right) + A\left(s, a; \theta, w^A\right) \quad (7)$$

The neural network carries out the initial judgment of the data. Then it completes the action reward correction so that the output action is more aligned with the actual situation. The target value of the D3QN model is given by

$$Y^{D3QN} = E_{(s,a,r,s')}\left[r + \gamma Q_{\mathrm{T}}\left(s', \operatorname{argmax} Q_{\mathrm{M}} \times \left(s', a'; \theta, w^V, w^A\right); \theta', w^V, w^A\right)\right] \quad (8)$$

The loss function for updating the network parameters is denoted as

$$L\left(\theta, w^V, w^A\right) = E_{(s,a,r,s')}\left[Y^{D3QN} - Q_{\mathrm{M}} \times \left(s', a'; \theta, w^V, w^A\right)^2\right] \quad (9)$$

III. D3QN-BASED RADAR JAMMING DECISION TECHNOLOGY IMPLEMENTATION

A. D3QN-BASED RADAR JAMMING DECISION MODEL

In order to reasonably simplify the problem and highlight the keypoint of the radar jamming decision process, this paper does not consider the specific equipment types, operator man-made errors, and other influencing factors. The D3QN model includes the following four elements.

(1) State-space S. The set of states represents the operating modes of multifunctional radar. For example, phased array radar has many modes, such as detection, tracking, guidance, and measurement parameters.
(2) Action space A. Action space is noted as a set of jamming strategies that the jamming party use in the electronic countermeasures. For example, the jammer has

deceptive jamming, suppression jamming, and other jamming patterns.

(3) Transfer probability function $P\left(s'|s,a\right)$. It denotes the probability that the jammer changes the radar's state to s' by jamming action a when the radar operating state is s.

(4) Reward function $R\left(s,a\right)$. It indicates the immediate return value after taking a particular jamming action. R values are defined by the change of radar threat level after jamming, we set
   a) $R = 100$, the state of radar switches to a lower threat level.
   b) $R = 0$, no transformation of the radar threat level.
   c) $R = -100$, the state of radar switches to a higher threat level.

When the state of multifunctional radar is s at the moment, the jammer selects a jamming pattern according to the ε-*greedy* strategy. We analyze the change of radar threat level after jamming and store the obtained sample data $\left(s,a,r,s'\right)$ in the experience pool. The experience pool is a memory playback unit to store the experience samples obtained from the jamming countermeasure. The neural network is updated with randomly selected samples from the unit during training. The data sampling follows the prioritized experience replay mechanism. We calculate the action reward value using the estimation value network and update the network parameters with the mean squared difference as the loss function. The target value network outputs Y as the final reward value. After sufficient training and learning in the adversarial environment, the optimal jamming strategy can be output when the cumulative reward values converge. The flow chart of the D3QN for multifunctional radar jamming decisions is as follows.

The above method realizes an autonomous online closed-loop learning process, effectively improving the countermeasure level of jamming decision models. It meets the requirements of intelligent, dynamic, and real-time cognitive electronic warfare.

B. NETWORK STRUCTURE

Since the state of the multifunctional radar are high-dimensional continuous, the discrete state space increases the difficulty of the decision process. Therefore, the D3QN uses the nonlinear fitting capability of the Dueling network to obtain a more accurate estimation value network function. As a result, the jammer can better reduce the action-value error after completing jamming for different radar states.

We input radar states s in the Dueling network, and output the state value function $V\left(s;\theta,w^V\right)$ and action advantage function $A\left(s,a;\theta,w^A\right)$ respectively after the hidden layer processing.

The state value function represents the value of the radar threat level change after jamming. The action advantage function represents the value by choosing a particular jamming action and outputs a vector of dimension |A|. Then, the state value function and the action advantage function are adopted to do linear fitting to obtain the real reward value $Q_{\mathrm{M}}\left(s,a;\theta,w^V,w^A\right)$ of each jamming pattern

$$Q_{\mathrm{M}}\left(s,a;\theta,w^V,w^A\right) = V\left(s;\theta,w^V\right) + A\left(s,a;\theta,w^A\right) - \frac{1}{|A|}\sum_{a'\in A} A\left(s,a';\theta,w^A\right) \quad (10)$$

This paper uses a forward 3-layer fully connected neural network to fit the action value approximately. The neural network structure is shown in Figure 3.

C. PRIORITY EXPERIENCE REPLAY MECHANISM

The premise of neural network training assumes that the training data are independent and identically distributed. However, the jammer can only get the reward value by observing the state change of the target radar. This situation leads to a correlation between the interaction data and does not meet the neural network training conditions. Therefore, DQN adopts the "Experience Replay" mechanism and randomly selects samples to update the network, solving the distribution problem caused by the correlation data [32], [33].

In the actual radar jamming decision, the random sampling method tends to ignore the differences between experience samples, resulting in sampling inefficiencies and increasing the decision time consumption. Therefore, this paper proposes an improved prioritized experience replay mechanism based on temporal difference error (TD-error) [34]. The TD algorithm uses the value of the difference between the target and estimated Q value to evaluate the priority of samples

$$\varepsilon = \left|R + \gamma \max Q\left(s_{\mathrm{t}+1},a_{\mathrm{t}}\right) - Q\left(s_{\mathrm{t}},a_{\mathrm{t}}\right)\right| \quad (11)$$

Then ε is selected as temporal difference error in the D3QN network. It indicates that learning this sample can make the network obtain a better improvement effect, and its priority $I(i)$ should be higher. The sampling priority $I(i)$ is given by

$$I(i) = \frac{I_i^{\alpha}}{\sum_n I_n^{\alpha}} \quad (12)$$

where i is the sample serial number. However, the jammer will often visit samples with larger absolute values and rarely or not visit some samples, leading to local convergence of the strategy, which is difficult to provide reliable guidance for the actual jamming decision process. Therefore, this paper intends to assign a higher sampling priority to the experience samples with low access frequency. The state distribution of different experience samples is given by

$$I(m) = \int_i^M \sum_{\mathrm{i}=0}^{\infty} \gamma^i I(m_0) I\left(m_{\mathrm{t}} \to m_{\mathrm{t}+1}\right) d_{m_{\mathrm{t}}} \quad (13)$$

where $I(m_0)$ is the probability of the initial state. When the decision process proceeds, if the sampling probability $I(m)$ of sample m_{t} is large, implying that the jammer often updates the neural network using the same radar state. So it is appropriate to reduce the sampling frequency of the experience sample m_{t}. Then, more samples update the neural network to maximize the information value of each radar

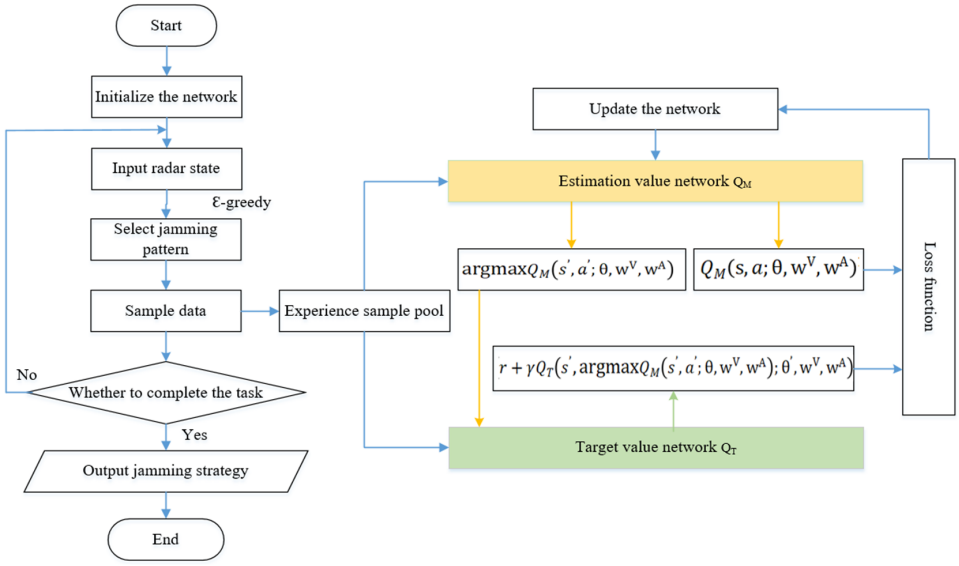

FIGURE 2. Flow chart of D3QN-based multifunctional radar decision-making method.

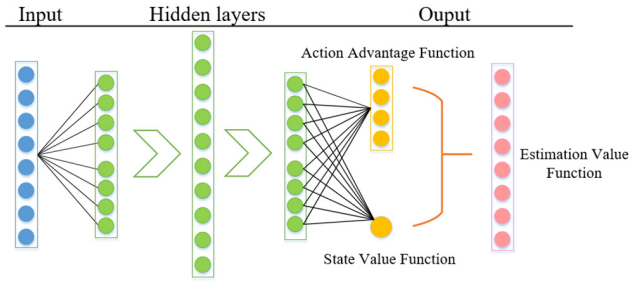

FIGURE 3. Network structure.

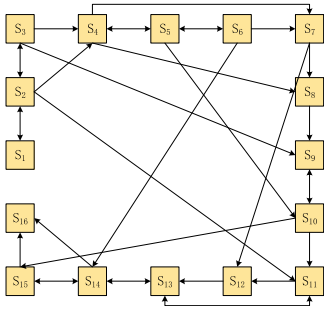

FIGURE 4. Radar state transition network.

state, which can effectively improve the decision efficiency and reduce the impact of the local optimum on the decision accuracy.

IV. SIMULATION VERIFICATION

A. DESCRIPTION OF THE COUNTERMEASURES ENVIRONMENT

Multifunctional radar generally has a variety of operating states. In the actual countermeasure process, the radar threat level is gradually reduced by the jamming. For example, when a multifunctional radar is in the guidance state, the radar may lose some parameter information after jamming, the radar can not lock on the target continuously. Thus, the radar only shifts to the imaging state with lower threat levels. The imaging accuracy and precision of the radar decrease by continued jamming. As a result, the radar can not detect the target and transforms it into the coarse search state. This situation can be considered that the effect of the jamming process is significant. Therefore, radar generally does not switch from the highest known threat level to the lowest threat level [35].

We completed the experiments in a Matlab environment with the experimental platform parameters of Intel(R) Core(TM) i7-10750H CPU@2.60 GHz processor, 16G RAM, and no graphics acceleration is used.

We assume a multifunctional radar has sixteen radar operating states $S_{sample} = \{s_1, s_2, s_3 \ldots, s_{16}\}$ and the jammer can take nine jamming patterns $A = \{a_1, a_2 \ldots, a_9\}$. Then we generate a connected network with random transformation relationships using Matlab. Figure 4 shows that the network nodes represent the radar states and the arrow lines between the nodes indicate the state transition direction. We define the highest threat level for the state s_1 of the radar, the target state s_{16} with the lowest threat level. The transfer probabilities P_t between states conform to a Gaussian distribution with the mean value of μ and the variance is σ^2, and $P_t \in [0, 1]$ indicates that the probability of transferring a radar state to other radar states, which sums to 1 [17], [18], [20].

The neural network built for training is a 3-layer, fully connected layer. The number of nodes in the input layer is the radar state dimension. The number of nodes in the output layer is the jamming pattern dimension. Finally, the intermediate layer is connected to the Dueling network, and other parameters are set in Table 1.

B. SIMULATION PROCESS

We first initialize the network parameters before the start of the jamming decision process. Then we extract 10% of the samples with lower sampling frequencies from the experience

TABLE 1. Parameters table.

Parameters	Value
number of training rounds	2000
total experience pool	10000
number of intermediate layer nodes	128
network update rate	100 rounds
exploration factor	$1 \rightarrow 0.2(0.1)$
learning rate	0.01
reward discount factor	0.99
transfer probabilities	P_t
reward function	R
number of radar states	16
number of jamming patterns	9

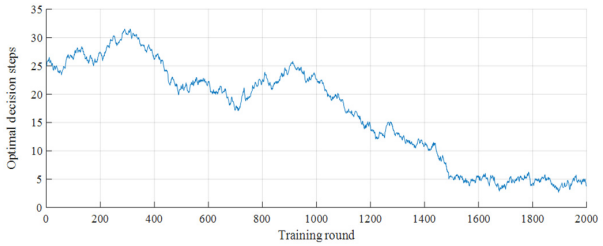

FIGURE 5. Decision-making results of the D3QN method.

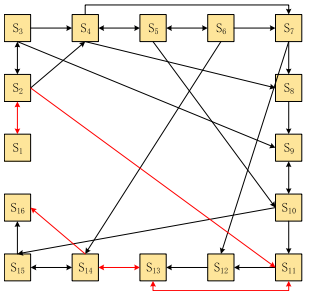

FIGURE 6. Optimal decision route.

pool for calculating the loss function. Finally, we update the estimation value network Q_{M} according to the calculation results and replace the parameters θ' with the current θ every 100 training rounds.

The radar state starts from s_1, and the transition ends at s_{16}. The jamming that makes the fastest transition is considered the optimal strategy.

In order to use all the jamming patterns, the exploration factor is initially set to 1. The jamming pattern is randomly selected at the beginning of the decision process. The exploration factor decreases by 0.1 with every 100 training rounds and remains constant when it decreases to 0.2. The searching probability is only 20% at this moment, indicating that the jammer can take full advantage of the acquired experience at the end of training. The decision results are shown in Figure 5.

As can be seen from Figure 5, the horizontal coordinate represents the number of training rounds, and the vertical coordinate represents the decision steps. In the beginning of the countermeasure, due to the low experience in the network, the jammer can only explore through many aimless attempts. As the number of interactions increasing, the jammer stores the learned experience in the experience sample pool. The neural network introduces the priority

TABLE 2. Cumulative reward value.

Round	DQN	DDQN	D3QN-ER	D3QN-PER
0-200	126.7	389.2	-229.4	31.4
200-400	401.2	613.0	-288.3	239.0
400-600	587.5	207.9	198.2	465.1
600-800	398.2	441.2	781.5	1075.3
800-1000	177.9	704.6	1282.1	1280.6
1000-1200	203.6	1190.5	1352.0	1432.3
1200-1400	368.1	1239.3	971.0	1391.4
1400-1600	249.5	1140.8	928.7	1434.1
1600-1800	417.9	1066.7	1194.9	1533.7
1800-2000	782.4	852.6	1301.9	1643.9

experience replay mechanism and significantly reduces the number of decision steps. As a result, the learning efficiency is sharply improved. Eventually, the training reaches a steady state at about 1500 rounds.

Furthermore, the decision curve finally converges in about five steps, coinciding with the minimum number of steps required in the network constructed in Figure 6. This indicates that the jammer has learned the best jamming strategy. The jammer uses less a priori knowledge and ultimately completes the decision task.

C. COMPARATIVE ANALYSIS OF METHODS

The D3QN-based multifunctional radar decision-making method introduces the DDQN [24] to improve the decision accuracy. Furthermore, we improve the sample utilization and shorten the decision time through the prioritized experience replay mechanism [20].

Generally, the more cumulative reward values the jammer obtains when training, the more times the jammer can successfully transition during the jamming decision process. Therefore, we demonstrate the improvement effect of the overall decision by analyzing each part. We simulate and compare the cumulative reward value for 2000 rounds of only a single metric. The results of the cumulative reward values are shown in Figure 7. Where the horizontal coordinate of the curve represents the total decision rounds, the vertical coordinate represents the cumulative reward value in a decision round. The D3QN-ER and D3QN-PER represent the methods introduced the experience replay mechanism of the literature [34] and this paper, respectively. The corresponding average reward values per 200 rounds are recorded in Table 2.

Figure 7 and Table 2 show that although all four methods can maximize their cumulative reward value, the DQN method has a poor convergence effect due to the overestimation problem. The cumulative reward value is only taken to the maximum value of 782.4 in the 1800 to 2000 rounds, which is difficult to provide reliable help to the jammer.

DDQN uses a different network structure and effectively avoids the influence of the local optimum on the decision. The maximum value is 1239.3 from 1200 to 1400 rounds, which improves the decision effect compared with the DQN method. However, the deviation in the Q-value calculation will gradually increase, and the final effect at the end of training is unsatisfactory.

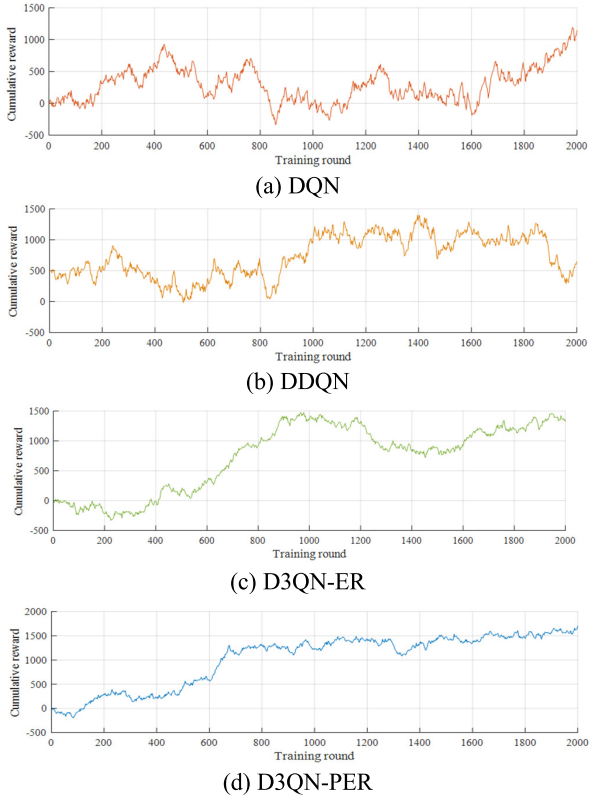

FIGURE 7. Cumulative reward value.

The D3QN-ER method samples by the experience replay mechanism based on temporal difference error (TD-error) [34]. D3QN-ER makes the Q-value calculation more accurate, with a maximum value of 1352.0 from 1000 to 1200 rounds. It has a particular enhancement effect on the decision-making process. However, repeated sampling also reduces the valuable information in the subsequent training process and an inevitable magnitude decrease in the gain value.

The D3QN-PER maximizes the information of the samples by using the prioritized experience replay mechanism proposed in this paper. It combines the advantages of other methods to make the final reward value curve relatively smooth. The reward value converges at rounds 600 to 800, and the maximum reward value is 1643.9, which is approximately 2.1 times higher than the DQN method. This shows that the method obtains the best jamming decision scheme with less training times, avoiding the waste of jamming resources and making the decision process more effective and stable.

When the decision accuracy converges, it can be considered that the method has learned the optimal strategy, and the overall decision result will not change over time. Therefore, we use the simulation environment designed in Section A to compare the methods in this paper with the current main methods [18], [19]. We define the percentage of times each method makes the successful transition in 2000 rounds as the decision success rate. Then we record the decision time as an index to evaluate the efficiency of these methods. The results are shown in Figure 8.

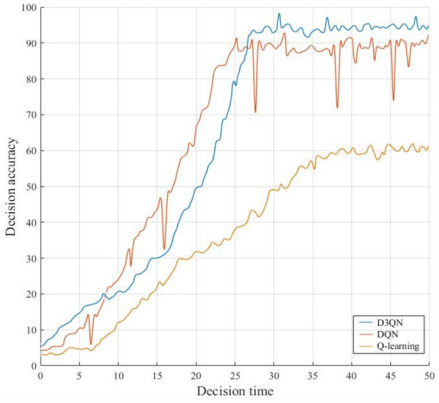

FIGURE 8. Performance comparison results of three decision methods.

When the decision process is stabilized, the Q-learning method needs to establish a large-scale state action table, which leads to many calculations and further prolongs the decision time, and finally takes about 37s. As a result, the decision accuracy of this method is only about 60%, which is challenging to complete the radar jamming decision task in real-time and accurately.

The DQN method introduces neural networks in calculating Q values, effectively avoids the problem of increasing the number of decision dimensions. As a result, the overall decision process is more efficient and only takes approximately 25s to arrive at the optimal decision method. However, the same structure function in calculating the Q value leading to a frequent overestimation of the Q value. As a result, even when the decision accuracy tends to be stable, the system still has the probability of choosing the suboptimal strategy, which results in a significant decision error. It will mislead the jammer to choose the wrong jamming action, delay the best jamming time.

Although the D3QN method takes slightly longer to converge the decision accuracy than the DQN method, it effectively reduces the calculation error and the impact of the overestimation problem on the decision result. The overall change of the decision accuracy curve more stable. Therefore, D3QN can provide a more reliable decision-making process for the jammer and has better practical application value. In summary, the multifunctional radar cognitive jamming decision method based on D3QN has achieved better results.

V. CONCLUSION

In this paper, we solve the slow convergence and Q-value overestimation problems of existing DQN-based radar jamming decision methods. Firstly, we build the decision model according to the multifunctional radar countermeasure process. Then, the action selection and effectiveness evaluation are generated with different functions. Finally, the prioritized experience replay mechanism is used further to improve the training efficiency of the neural network and shorten the optimal decision time. The simulation experimental result shows that the D3QN method is more stable and reliable. The D3QN completes decision tasks 2.1 times more efficiently

than DQN and improves decision accuracy by approximately 10% over DQN. In whole, the D3QN method can be used as an effective method of the multifunctional radar jamming decision technology, which lays a good foundation for the engineering implementation of cognitive electronic warfare systems.

REFERENCES

[1] S. Haykin, "Cognitive radar: A way of the future," *IEEE Signal Process. Mag.*, vol. 23, no. 1, pp. 30–40, Jan. 2006.

[2] S. Johnston, "Radar electronic counter-countermeasures," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-14, no. 1, pp. 109–117, Jan. 1978.

[3] F. A. Butt and M. Jalil, "An overview of electronic warfare in radar systems," in *Proc. Int. Conf. Technol. Adv. Electr., Electron. Comput. Eng. (TAEECE)*, May 2013, pp. 213–217.

[4] D. J. Bachmann, R. J. Evans, and B. Moran, "Game theoretic analysis of adaptive radar jamming," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 1081–1110, Apr. 2011.

[5] B. Goertzel, "Artificial general intelligence: Concept, state of the art, and future prospects," *J. Artif. Gen. Intell.*, vol. 5, no. 1, pp. 1–46, Mar. 2014.

[6] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behav. Brain Sci.*, vol. 40, pp. 1–72, Nov. 2017.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, pp. 1–9, Dec. 2013.

[8] V. Mnih, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[9] Y. Shin, J. Kim, K. Jin, and Y. B. Kim, "Playtesting in match 3 game using strategic plays via reinforcement learning," *IEEE Access*, vol. 8, pp. 51593–51600, 2020.

[10] C. Holmgard, M. C. Green, A. Liapis, and J. Togelius, "Automated playtesting with procedural personas through MCTS with evolved heuristics," *IEEE Trans. Games*, vol. 11, no. 4, pp. 352–362, Dec. 2019.

[11] D.-J. Yang, X.-X. Qin, X.-D. Xu, C.-S. Li, and W. Guo, "Sample efficient reinforcement learning method via high efficient episodic memory," *IEEE Access*, vol. 8, pp. 129274–129284, 2020.

[12] L.-H. Lu, S.-J. Zhang, D.-R. Ding, and Y.-X. Wang, "Path planning via an improved DQN-based learning policy," *IEEE Access*, vol. 7, pp. 67319–67330, 2019.

[13] C. Shyalika, T. Silva, and A. Karunananda, "Reinforcement learning in dynamic task scheduling: A review," *Social Netw. Comput. Sci.*, vol. 1, no. 6, pp. 1–17, Sep. 2020.

[14] S. Kang, H. Park, S. Noh, S. R. Park, K. Kim, S. Lyu, and S. Kim, "Autonomously deciding countermeasures against threats in electronic warfare settings," in *Proc. Int. Conf. Complex, Intell. Softw. Intensive Syst.*, Mar. 2009, pp. 177–184.

[15] Y. Zhang, G.-Y. Si, and Y.-Z. Wang, "Modelling and simulation of cognitive electronic attack under the condition of system-of-systems combat," *Defence Sci. J.*, vol. 70, no. 2 pp. 183–189, Mar. 2020.

[16] Y.-J. Li, Y.-P. Zhu, and M.-G. Gao, "Design of cognitive radar jamming based on *Q*-learning algorithm," *Trans. Beijing Inst. Technol.*, vol. 35, no. 11, pp. 1194–1199, Nov. 2015.

[17] Q. Xin, W.-G. Zhu, and X. Jia, "Intelligent countermeasure design of radar working-modes unknown," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Oct. 2017, pp. 1–5.

[18] Q. Xin, W.-G. Zhu, and X. Jia, "Research on method of intelligent radar confrontation based on reinforcement learning," in *Proc. IEEE Int. Conf. Comput. Intell. Appl. (ICCIA)*, Dec. 2017, pp. 471–475.

[19] H.-Q. Li, Y.-L. Li, C. He, J.-W. Zhan, and H. Zhang, "Cognitive electronic jamming decision-making method based on improved *Q*-learning algorithm," *Int. J. Aeros. Eng.*, vol. 2021, Dec. 2021.

[20] B.-K. Zhang and W.-G. Zhu, "Research on decision-making system of cognitive jamming against multifunctional radar," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Aug. 2019, pp. 1–6.

[21] L. Gao, L. Liu, Y. Cao, S.-Y. Wang, and S.-X. You, "Performance analysis of one-step prediction-based cognitive jamming in jammer-radar countermeasure model," *J. Eng.*, vol. 2019, no. 21, pp. 7958–7961, Sep. 2019.

[22] F. Smits, A. Huizing, W. van Rossum, and P. Hiemstra, "A cognitive radar network: Architecture and application to multiplatform radar management," in *Proc. Eur. Radar Conf.*, Oct. 2008, pp. 312–315.

[23] W. Pan, X. Jin, H.-X. Xie, and Y. Xia, "Radar jamming strategy allocation algorithm based on improved chaos genetic algorithm," in *Proc. IEEE Chin. Control Decis. Conf. (CCDC)*, Aug. 2020, pp. 4478–4483.

[24] H.-D. Liu, H.-T. Zhang, Y. He, and Y. Sun, "Jamming strategy optimization through dual *Q*-learning model against adaptive radar," *Sensors*, vol. 22, no. 1, p. 145, Dec. 2021.

[25] 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, Jan. 1996.

[26] E. Akanksha, J. Sehgal, N. Sharma, and K. Gulati, "Review on reinforcement learning, research evolution and scope of application," in *Proc. 5th Int. Conf. Comput. Methodol. Commun. (ICCMC)*, Apr. 2021, pp. 1416–1423.

[27] J.-W. Li, W. Monroe, A. Ritter, M. Galley, J.-F. Gao, and D. Jurafsky, "Deep reinforcement learning for dialogue generation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (ACL)*, Nov. 2016, pp. 1–5.

[28] I. Osband, C. Blundell, A. Pritzel, and B. V. Roy, "Deep exploration via bootstrapped DQN," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2016, p. 29.

[29] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[30] H. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double *Q*-learning," in *Proc. Conf. Artif. Intell.*, Mar. 2016, vol. 30, no. 1, pp. 2094–2100.

[31] Z.-Y. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 48, Jun. 2016, pp. 1995–2003.

[32] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2016, pp. 34–39.

[33] S.-T. Zhang and R. S. Sutton, "A deeper look at experience replay," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Dec. 2018, pp. 4–9.

[34] Y.-N. Hou, L.-F. Liu, Q. Wei, X.-D. Xu, and C.-L. Chen, "A novel DDPG method with prioritized experience replay," in *Proc. IEEE Int. Conf. Syst., Man Cebern. (SMC)*, Oct. 2017, pp. 316–321.

[35] N.-J. Li and Y.-T. Zhang, "A survey of radar ECM and ECCM," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 31, no. 3, pp. 1110–1120, Jul. 1995.

LU-WEI FENG was born in 1998. He received the B.S. degree in electronic information engineering from Nanchang University, Nanchang, China, in 2020. He is currently pursuing the M.S. degree in electronic information engineering with Dalian Naval Academy. His research interests include electronic countermeasure and artificial intelligence.

SONG-TAO LIU was born in 1978. He received the B.S. degree in aviation radar and the M.S. and Ph.D. degrees in signal and information processing from the Naval Aeronautical Engineering Institute, in 2000, 2003, and 2006, respectively. He joined the Dalian Naval Academy, in 2006, where he is currently an Associate Professor with the Department of Information System. He has published over 100 papers in journals and conference proceedings. His research interests include electronic countermeasure, image processing, and optoelectronic engineering.

HUA-ZHI XU was born in 1988. He received the B.S. degree in military oceanography from Dalian Naval Academy, Dalian, China, in 2010, where he is currently pursuing the M.S. degree in electronic information engineering. His research interests include electronic countermeasure and operational effectiveness evaluation.