BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **23**, No 4
Sofia • 2023 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2023-0040

A Novel Self-Exploration Scheme for Learning Optimal Policies against Dynamic Jamming Attacks in Cognitive Radio Networks

Sudha Y.[1], *Sarasvathi V.*[2]

[1]*Department of Computer Science and Engineering, Presidency University, Bangalore, Karnataka, India.*
[2]*Department of Computer Science and Engineering, PES University, Bangalore, Karnataka, India.*
E-mails: sudhasohan@gmail.com / sudha.y@presidencyuniversity.in sarsvathiv@pes.edu

***Abstract*:** *Cognitive Radio Networks* (*CRNs*) *present a compelling possibility to enable secondary users to take advantage of unused frequency bands in constrained spectrum resources. However, the network is vulnerable to a wide range of jamming attacks, which adversely affect its performance. Several countermeasures proposed in the literature require prior knowledge of the communication network and jamming strategy that are computationally intensive. These solutions may not be suitable for many real-world critical applications of the Internet of Things* (*IoT*). *Therefore, a novel self-exploration approach based on deep reinforcement learning is proposed to learn an optimal policy against dynamic attacks in CRN-based IoT applications. This method reduces computational complexity, without prior knowledge of the communication network or jamming strategy. A simulation of the proposed scheme eliminates interference effectively, consumes less power, and has a better Signal-to-Noise Ratio* (*SNR*) *than other algorithms. A platform-agnostic and efficient anti-jamming solution is provided to improve CRN's performance when jamming occurs.*

***Keywords*:** *Cognitive radio network, IoT, Jamming attack, Customized environment, Self exploration, Reinforcement learning.*

1. Introduction

With the emergence of 5G technology, IoT devices can now leverage higher bandwidth, lower latency, and increased reliability, enabling various new applications and services [1]. However, the increased number of devices and the need for higher data rates require efficient utilization of the spectrum resources [2]. Cognitive Radio Network (CRN) can address these challenges by dynamically allocating and managing the spectrum resources to IoT devices based on their requirements and availability [3-5].

However, the open nature of CRN and IoT makes them vulnerable to different types of attacks, including jamming attacks [6, 7]. Jamming attacks can significantly degrade the quality of communication and even render the network unusable, thereby

compromising critical operations [8]. Jamming attacks can take various forms, such as Continuous Wave (CW) jamming [9], random pulse jamming [10], and sweep jamming [11]. Attackers now use sophisticated jammers as a result of technological advancements that target particular channels that are challenging to identify due to their extremely unpredictable and dynamic character [12-13]. These jamming attacks can have severe consequences in CRN-critical IoT applications. For example, in healthcare monitoring applications, jamming attacks can prevent the transmission of vital signs data, leading to delayed medical treatment and potentially life-threatening situations. The adversary can also use jamming to interfere with the spectrum sensing process and provide false information to the cognitive radio devices.

Among the traditional anti-jamming strategies that have been advocated for countering jamming attacks are frequency hopping [14], spread spectrum [15], and power management [16]. However, these strategies do not work against dynamic and sophisticated jammers, such as smart jammers. To combat intelligent jamming attacks, modern Machine Learning (ML) techniques have become increasingly popular with the rise of Artificial Intelligence (AI) [17]. The use of these methodologies enables countering jamming attempts and adapting to changing network conditions. While ML-based solutions can detect and counter dynamic jammers, there are many drawbacks, including high computational complexity and the need for large amounts of labeled training data. It has been shown that Reinforcement Learning (RL) techniques like Q-learning [18] and Deep Q-Network (DQN) generate effective anti-jamming rules [19]. For anti-jamming solutions based on Q-learning, extending the Q-table size causes computational overhead, and DQN's slow learning rate makes it appropriate for low-dimensional, discrete action spaces. In order to overcome these disadvantages, more effective anti-jamming technology is needed.

The proposed research manuscript presents a new self-exploration approach for developing optimal policies to defend against dynamic jamming attacks in CRN. The approach employs the Deep Deterministic Policy Gradient (DDPG) algorithm to train an anti-jamming agent that can adapt to fluctuating network conditions and effectively counter-jamming attacks. The agent learns from its own experiences without the need for labeled training data, thereby reducing the computational complexity and improving the performance against dynamic and intelligent jammers. The contribution of our proposed work can be summarized as follows:

- This paper proposes a novel self-exploration scheme for learning optimal policies against dynamic jamming attacks in CRN using the DDPG algorithm.
- A suitable and benchmarked environment modeling is done to execute the proposed DDPG agent algorithm.
- The performance of the suggested anti-jamming solution is assessed against both sweep and smart jamming strategies.
- The proposed intelligent anti-jamming technique is evaluated by comparing its outcomes with other ML models, considering factors such as power consumption and SNR over the progressive time slot.

2. Related work

This section briefly discusses the existing works in the literature proposed by different researchers to address jamming attacks in CRN-driven networking applications.

An anti-jamming solution based on game theory and the Markov Decision Process (MDP) has been reported in the work of Singh and Trivedi [20] against random jamming attacks in CRN. In this work, an analysis is done to examine the variation in jammer power adjustment strategies, and then RL algorithm is implemented to un-jam the network. Nallarasan and Kottrsamy [21] have modeled the CRN jamming problem as intrusion detection and presented a deep auto-encoder-based solution to detect jamming attacks. However, this approach is limited only to attack detection and does not discuss the applicability of their approach in the case of an intelligent jammer capable of exploiting the dynamics of the environment. In response to this, Ibrahim et al. [22] discuss the strategies adopted by smart jammers and present a solution using the principle of MDP. In this work, the authors have employed the Q-learning technique to implement an agent as a solution against a smart jammer. However, the method presented in this work suffers from the scalability issue due to the fact that the Q-learning algorithm is subjected to a curse of dimensionality issue. Xiao et al. [23] have investigated the anti-jamming power control problem. The Stackelberg equilibrium is derived for learning the power control strategy against a jammer. This model consists of a transmitter, receiver, and jammer. Moreover, a Q-learning is used to explore the best jamming resistance. Sudha and Sarasvathi [24] have presented an effective anti-jamming solution based on RL against rule-based jamming attacks, and in their next work, they consider a case scenario of a smart jammer [25]. Based on the simulation process, the effectiveness of their presented scheme has been demonstrated.

Thien et al. [26] suggest a method for preventing jamming assaults in multi-channel CRN based on game theory. The authors utilize transfer learning and actor-critic neural networks to identify the optimal communication channel for the transmitter to avoid jammers on the communication links. However, it may be subjected to biased outcomes toward selecting a single optimal action for a similar kind of jamming pattern. Huang et al. [27] propose a channel-hopping-based jamming-free strategy that is designed to withstand different types of jamming attacks in a Cognitive Radio Network (CRN). This technology allows the system to hop between available channels without any pre-assignment role, which means that it does not require a fixed channel allocation plan.

Hanawal et al. [28] have developed a model in which the user and the jammer are viewed as two players in a zero-sum game. The concept tries to offer a robust defense against jamming assaults. Gao et al. [29] have adopted a bimodal game strategy that involves interaction between the transmitter and the jammer. However, this method requires prior knowledge of jamming strategy and communication models. An application of the dual Q-learning technique is used by Zhang et al. [30] to address joint channel and power minimization in a multi-user jamming-free environment. The cross-layer investigation of the cognitive radio-based jammers'

and CRN's anti-jamming capabilities is done by Cadeau et al. [31] using MDP. Table 1 summarizes the above-discussed literature to provide a quick insight to the readers.

Table 1. Summary of the above-discussed literatures

Citation	Problem context	Method	Remark
Singh and Trivedi [20], 2012	Anti-jamming in CRN	Reinforcement learning algorithms	–
Nallarasan and Kottursamy [21], 2021	High-level feature extraction	Autoencoder	Limited to attack detection and does not discuss the applicability of their approach in the case of an intelligent jammer capable of exploiting the dynamics of the environment
Ibrahim et al. [22], 2021	Combat intelligent jamming	MDP and Q-learning	May suffer from scalability issues due to the curse of dimensionality
Xiao et al. [23], 2015	Power optimization	Cooperative reinforcement learning	Need more optimization in computational complexity
Sudha and Sarasvathi [24], 2022	Combating rule-based jammer	RL	Effective against sweep jammer
Sudha and Sarasvathi [25], 2022	Mitigate intelligent jammer impact on CRN communication	Adversarial Learning Algorithm	Achieved optimal outcome regarding signal quality and power usage
Thien et al. [26], 2021	Combating anti-jamming in multi-channel CRN	Actor-critic learning framework	May be subjected to biased outcomes toward selecting a single optimal action for a similar kind of jamming pattern
Huang et al. [27], 2017	Computational efficiency	Channel-hopping-based strategy	Prone to dynamic jamming attacks
Hanawal et al. [28], 2016	Combating rule-based jammer	Zero-sum game model	Offers comprehensive security
Gao et al. [29], 2018	A trade-off in resource optimization and system performance	Bimodal game strategy	Requires prior knowledge of jamming strategy and communication models
Zhang et al. [30], 2022	Channel corporation and power optimization	Dual Q-learning	Not much suitable for continuous search pace
Cadeau et al. [31], 2014	Performance and security trade-off	MDP	Achieved good security features but at the cost of higher computational resources

Hence, it can be seen that several anti-jamming solutions have been proposed in the literature. However, most of them are rule-based and rely on predefined thresholds and heuristics. These solutions are not adaptive and cannot cope with unknown and dynamic jamming attacks. Besides, most of them focus on defending against a single type of jamming attack. Moreover, there is a lack of comprehensive evaluation of the performance of these solutions in terms of power consumption and SNR over different time slots. Therefore, there is a need for a comprehensive anti-

jamming solution that can defend against different types of jamming attacks and can be evaluated in terms of power consumption and SNR.

3. Proposed system

In the proposed system, first, a simulation Environment (Env) is developed using OpenAI Gym, where a DDPG Agent (Ag) learns to choose the optimal transmission power to minimize the impact of the jamming signal. The simulation Env mimics the scenario of communication with sweep and smart jamming attacks, where the jammer tries to disrupt communication by transmitting a jamming signal. Ag selects the best transmission power based on the current State (S^T) from the environment and a learned Policy (π) for initiating Action (A^T). Ag learns from its own experiences and does not require labeled training data, which reduces computational complexity and improves performance against dynamic and smart jamming attacks. The proposed solution aims to maximize the accumulated Reward R^W by finding the optimal channel for the Secondary User (U^S), while also ensuring that Primary Users (U^P) are not using the communication channels and Jammer (J) do not cause interference. The architecture of the proposed system is illustrated in Fig. 1.

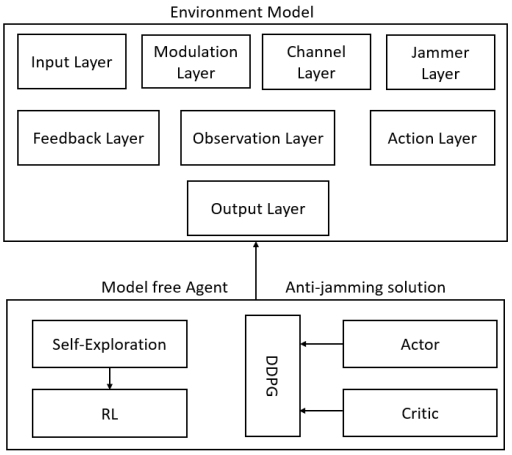

Fig. 1. Block-wise schematic architecture of the proposed system

The implementation of the proposed system consists of two main blocks: i) Env, and ii) value-based model free agent Ag as an anti-jamming solution. The modeling of the environment is done in a systematic manner using the OpenAI Gym function to imitate the CRN communication scenario with jamming attacks. The proposed system consists of a total of 8 layers. The first is the input layer, which receives the input parameters, which include the Channel information (Ci), Modulation (Md) scheme, and the presence of a Jammer (J). The second is modulation layer which performs the Md on the input data, using the BPSK modulation scheme. The third one is the channel layer which simulates the wireless channel by adding noise and fading to the modulated signal. The fourth jammer layer simulates the presence of the J, either as a sweep jammer or an intelligent jammer. A fifth layer receives the A^T

taken by the Ag, including frequency band selection and preference strategies. The sixth module is the observation layer that lets Ag to observe the $S^T$of the Env, including the jamming pattern, successful transmission rate, power consumption, and SNR. The seventh module is the feedback layer which provides feedback to the Ag based on the observed state, indicating the reward R^{W+}or penalty $R^{W-}$for the action taken and the final output layer provides the output of the Env, which includes the S^T and feedback.

The next block of the proposed system presents the design of a DDPG-based value-based intelligent and model-free Ag algorithm. The action taken by Ag is to select a frequency band that maximizes its successful transmission rate and avoids jamming signals. The $S^T$of the Env is the spectrum utilization and the presence of jamming signals. Ag gets $S^T \in$ Env and selects an A^T based on its current Policy (Pi). The Reward R^W factor is defined as the successful transmission rate minus a penalty for selecting a jammed frequency band. Here, Ag uses the DDPG algorithm to learn the optimal policy $\pi(S^T)$. The overall process can be mathematically expressed as follows:

Let's define the interaction between Ag and Env as a Markov Decision Process (MDP) defined by Tuple (T) such that: Ag $\rightleftharpoons$ Env $\leftarrow T = (S^T, A^T, P^T, R^W, \gamma)$. Here, P^T is the state transition probability function, and γ is the discount factor. The state of the environment S^T is a function of the past observation past, i.e., $s = f(h(t))$, where $h(t)$ represents the history of the past t observations. Ag takes an optimal A^{To} from the set of available A^T. It is to be noted here that A^T at time step t leads to a transition to a new state S^{T+1} with Probability $P(S^{T+1} | S^T, A^T)$. Therefore, $\forall$ A^T Ag $\leftarrow R^W(S^T, A^T)$, i.e., reward for taking an action $a \in A^T$ in state $s \in S^T$. The ultimate goal of Ag is to learn a $\pi(S^T)$ which capitalize on the expected R^W over the progressive time. The most suitable action policy $\pi(S^T)$ is determined by iteratively estimating *Q*-value function, given as follows:

(1) $$Q(S^T, A^T) = E[R^W(t+1) + \gamma \max Q(S^{T+1}, A^{T\prime}) \mid S^T, A^T].$$

Here, $E[.]$ denotes the expected value $\forall$ S^{T+1}, and $A^{T\prime}$ represents the next action taken by the Ag in state S^{T+1}. The algorithm updates the Q-value function iteratively using the Bellman optimality equation given as follows:

(2) $$Q(S^T, A^T) = Q(S^T, A^T) + \alpha[R^W(t+1) + \gamma \max (S^{T+1}, A^{T\prime}) - Q(S^T, A^T)],$$ where α is the learning rate.

Upon updating the value function, the algorithm estimates the $\pi(s)$ $\forall$ S^T. The policy $\pi(s)$ is defined as a deterministic mapping of observation from S^T to A^T, such that: $A^T = \mu(S^T)$. The algorithm updates the policy function iteratively using the following equation:

(3) $$Q(S^T, A^T) = R(S^T, A^T) + \gamma\, Q\left(S^{T+1}, \mu(S^{T+1})\right),$$

where S^{T+1} is the next state, and $\mu(S^{T+1})$ is the next action A^T taken by the agent Ag in state S^{T+1}. The agent uses the actor-critic method to estimate the optimal policy, where the actor estimates the policy function and the critic estimates the *Q*-value function.

3.1. Implementation of environment

The process of building the OpenAI Gym environment for the proposed work involves defining the observation space, action space, and reward function. Let us discuss each of them in detail.

- **Observation Space.** The observation space represents the state of the environment. In this case, the observation space includes the available frequency band and the presence of any jamming signals. We can represent the observation space as a vector:

(4) $$O_t = [f_1, f_2, \ldots, f_n, J_1, J_2, \ldots, J_n],$$

where f_i represents the availability of the *i*-th frequency band, and J_i represents the presence of a jamming signal in the *i*-th frequency band.

- **Action Space.** The action space represents the possible actions that the agent can take. In this case, the action space includes selecting a frequency band to transmit. We can represent the action space as a vector:

(5) $$\mathrm{A^T} = [a_1, a_2, \ldots, a_n],$$

where a_i represents whether the agent selects the *i*-th frequency band for transmission (a_i = 1) or not (a_i= 0).

- **Reward Function.** The reward function represents the goal of the agent. In this case, the goal is to maximize the successful transmission rate while minimizing power usage and avoiding jamming signals. We can represent the basis of deciding reward function as:

(6) $$\mathrm{R^W} = \mathrm{ST} - P_t - J_t,$$

where ST represents the Successful Transmission rate, P_t represents the Power usage, and J_t represents the presence of Jamming signals.

Considering the above basis, the Reward $\mathrm{R^w}$ for the agent in the proposed scheme can be numerically expressed as follows:

(7) $$\mathrm{R^W} = R_{\mathrm{SNR}}(\mathrm{A^T}) - c(\mathrm{A^T}),$$

where $R_{\mathrm{SNR}}(\mathrm{A^T})$ is the reward achieved by Ag for every successful transmissions, and $c(\mathrm{A^T})$ is the cost factor subjected communication quality associated with $\mathrm{A^T}$ taken by Ag for channel switching.

The criteria for transmission success are determined by the SNR factor at the receiver side. If the SNR of the received signal exceeds the demodulation threshold ($\mathrm{Md_{th}}$), the transmission is considered to be successful, and the reward $\mathrm{R^W}$ will be equal to 1. Otherwise, if the transmission fails, the Reward $\mathrm{R^W}$ will be –1, numerically given as follows:

(8) $$\mathrm{R^W} = \{(1 \text{ if } \mathrm{SNR\_}(\mathrm{R_x}) \geq \mathrm{SNR\_cutoff}, \text{ Otherwise } -1)\}.$$

$\mathrm{SNR_{R_x}}$ refers to the SNR of the received signal, which is used to compute the Reward $\mathrm{R^W}$. The proposed algorithm includes a control channel that is used to transmit signals temporarily to a Receiver, denoted by $\mathrm{R_x}$. This control channel is designed to be secure, which means that it cannot be compromised or affected by a jamming attack. By having a secure control channel, the system can ensure that critical information or commands are transmitted reliably and without interference from jammers, which could potentially disrupt or compromise the operation of the system. As already discussed, that the proposed study develops an environment using

functionalities provided by Open-AI Gym. The above-defined factors the observation space, action space, and reward function, the environment is then fit with Gym function to train and evaluate reinforcement learning algorithms for the anti-jamming mechanism.

3.2. Mathematical model for describing communication

Let there be N sub-bands available for transmission, each having a Bandwidth of B. Let M be the total number of SUs in the network. The environment can be modeled as a set of tuples $\text{Env} = \{S^T, A^T, P^T, R^W\}$. The communication model in the environment can be represented as follows:

- At each time step, the SUs explores the available sub-band channels to detect a spectrum hole.
- Once a spectrum hole is detected, the SU chooses an available channel to transmit its data.
- The Transmitter (Tx) modulates the data using BPSK and conveys it over the chosen channel.
- The Receiver (Rx) demodulates and receives the signal from the transmitter.
- The environment calculates the reward for the SU based on the SNR of the transmitted data and updates the state of the environment accordingly.
- The process continues for the next time step.

3.3. BPSK Modulation

BPSK stands for Binary Phase Shift Keying, which is a type of modulation scheme used in wireless communication systems. In the context of the proposed study, BPSK is considered because it is a relatively simple and efficient modulation scheme that can be used for wireless communication over Wi-Fi channels. BPSK only uses two signal levels to represent the transmitted data, which simplifies the hardware requirements and reduces power consumption. Additionally, BPSK is less susceptible to errors caused by noise and interference compared to other modulation schemes like QPSK or 16-QAM. This makes BPSK a good choice for reliable communication in wireless environments with noise and interference. Mathematically, the overall process of BPSK can be described as follows:

BPSK is a digital modulation scheme that represents binary 0 and 1 by shifting the phase of the carrier signal by 0 and π radians, respectively. Mathematically, the modulated signal can be represented as

$$s(t) = A * \cos(2\pi f_c t + \varphi), \tag{9}$$

where A is the Amplitude of the carrier signal, f_c is the frequency of the carrier signal, t is time, and φ is the phase shift. Here, the phase shift φ can take two values: 0 and π. Let the signal rate be R_s (i.e., the number of signals transmitted per second), and let the bit Rate be R_b (i.e., the number of bits transmitted per one second). Then, the relationship between the signal rate and the bit rate can be given as

$$R_b = R_s * \log_2(M), \tag{10}$$

where M is the number of signals used in the modulation. For BPSK, $M = 2$, and therefore, $R_b = R_s$.

Let the message signal to be transmitted be a binary sequence of bits $\{b_k\}$ of length *N*. The modulated signal for BPSK can be expressed as:

for bit value b_k = 1,

(11) $$s(t) = A * \cos\big(2\pi f_c t + \pi(1 - b_k)\big);$$

for bit value $b_k = 0$,

(12) $$s(t) = A * \cos(2\pi f_c t + \pi b_k).$$

At the receiver side, the obtained signal is demodulated by multiplying it with a locally generated carrier signal that is synchronized with the transmitted carrier signal. The received signal can be expressed as

(13) $$r(t) = s(t) * \cos(2\pi f_c\, t + \theta) + n(t),$$

where θ is the phase difference between the transmitted and received carrier signals, and $n(t)$ is the Gaussian noise with zero mean and variance $N_0/2$.

The received signal is then passed through a matched filter to obtain the baseband signal, which is compared with a threshold value to decide the transmitted bit value. The decision can be made as

(14) $$b_k = 1 \text{ if } r(t) > 0,\ b_k = 0 \text{ if } r(t) < 0.$$

3.4. Self-Exploration based on Model-free agent

Our proposed intelligent anti-jammer DDPG agent is trained to make decisions based on its past experiences to minimize the impact of sweep jammers. The agent's training process involves repeatedly interacting with the environment, observing the state, taking action, receiving a reward, and updating its neural network to learn the optimal policy.

- **Example scenario.** A sweep jammer is transmitting a jamming signal in a frequency sweep from 2.4 GHz to 2.5 GHz, and the agent is trying to find a jamming-free channel to communicate. The agent starts by exploring the available channels and sensing the environment's state. It then uses its neural network to determine the best action to take, which is to switch to a different channel. The agent takes this action and receives a positive reward, as the communication is now jamming-free. The agent continues to monitor the environment and takes appropriate actions to maintain the communication link. The DDPG algorithm is employed to train the agent, which is a value-based, model-free actor-critic neural network. The actor-network is responsible for learning how to choose an action based on the current state, while the critic network assesses the effectiveness of the selected action. The agent learns from experience, which is stored in a replay buffer and used for training. Once the agent is trained, it can be deployed in a real-time environment to defend against sweep jammer. The agent observes the current state of the environment, selects an action and receives a reward based on its action. The agent continues to learn and improve based on its experience in the real-time environment.

The algorithmic steps for implementing the proposed anti-jamming solution are discussed as follows:

- The input parameters to the algorithm are:

i. *E* (Episodes): The number of times the agent will go through the training process.

ii. *I* (Iterations): The number of iterations to be performed in each episode.

iii. α (Learning rate): The rate at which the agent learns from its experiences.

iv. γ (Discount): The discount factor used to discount future rewards.

- The output of the algorithm is the selection of a jamming-free signal.
- The algorithm initializes random weights (w) for the current actor (μ) and critic (Q) network as well for target networks. The algorithm then starts the training process, which consists of going through multiple episodes.
- For each episode, the algorithm initializes the state of the system (ξ) and starts iterating through the process.
- The algorithm chooses an action (a^u) for the user based on the current state.

Algorithm 5. Anit-Jammer

Initialize random weights θ for the current network: [μ(actor), Q(critic)]
Initialize random weights θ' for target network: [μ'(actor), Q(criitc)$'$]
Initialize memory buffer *b*
Set exploration rate *r*
For all State *S* and user Action *A* and Jammer action *J*
Set state to initial observation
For each episode $=$ 1: E_P do
  For Iteration in range *I*:
  Choose action a: $\mu(s)$ + $E \times N(0, 1)$
Perform frequency hopping
Sense frequency spectrum
Select channel *i* as per current policy
Action a^u→ initiate transmission
Get reward $R(s, a^u)$
Get next state (s+1)
Get Tuple $[s, a^u, R, s+1]$ into experience pool ψ
If $|\psi| \geq B$ **do**
Randomly select *B* from ψ
Endif
For each experience in *B* **do**
Compute value for μ' and Q'
Update $Q_\theta(s, a^u)$by minimizing the loss $\mathcal{L}(\theta)$
Update the $\mu(s)$based on policy gradient using
Update μ' and Q'
End for
$s \leftarrow s+1$ and $a^u \leftarrow a^{u+1}$
End while
If $a = a^j$ **then**
Signal jammed, and Smart Jammer gets reward
else
Transmission successful, agent rewarded
End if
End

- The algorithm further performs frequency hopping to perceive the communication channel. The algorithm also selects channel (i) from the available Channels (C) and the Primary user according to the current policy.
- The algorithm initiates the transmission by selecting action (a^u). Then the algorithm calculates the Reward factor (R) for the current state and action using the equation (1 if $B(f_u) \cong B(f_j)$: True Otherwise: 0).
- The algorithm then moves to the next state (s+1) and adds the tuple $[s, a^u, R, s+1]$ to the experience pool (ψ). If the number of experiences in the pool is greater than or equal to the Batch size (B), the algorithm randomly selects a batch of experiences from the pool.
- For each experience in the batch, the algorithm computes the target value (y_k) using the current and target networks. The algorithm then updates the Q-value function (Q_θ) using the loss function. The algorithm then updates the actor (μ) based on the policy gradient using the equation. The algorithm then updates the target networks (μ' and Q') using the soft update rule. The algorithm checks whether the selected action (a) is the same as the jammer's action (a^j). If the signal is jammed, the jammer gets a reward. Otherwise, the transmission is successful, and the proposed agent gets a positive reward.
- The algorithm then moves to the next state (s+1) and the next action ($a^u + 1$) to start the next iteration. The algorithm repeats steps until the end of the episode. The algorithm repeats steps for each episode until the end of the training process. Finally, the algorithm outputs the selection of a jamming-free signal.

4. Implementation and experiments

In this section, the results of applying the proposed self-exploration scheme against sweep jamming and smart jamming attacks are presented. The performance evaluation is based on the power consumption and SNR. The proposed anti-jamming solution is developed using Python and executed in an Anaconda development environment installed on Windows 10 Intel i7. Table. 2 presents the simulation parameters used in the proposed system.

Table. 2. Simulation parameters

Parameters	Value
WiFi Frequency band	2.4 GHz
Number of communication channels	11
Jamming model	Sweep jammer
Jamming power	30 dB. m
Transmitter signal power	$25 - 45$ dB. m
Bandwidth of the Transmitter signal	20 MHz
Bandwidth of the Jamming signal	20 MHz
Demodulation cut-off	10 dB
Data rate	2 Mbps
Digital modulation technique	BPSK
Channel switching Cost	0.2
Discount factor	0.96
Minibatch size	32

The experimental setup involves a communication scenario with a transmitter and receiver (Tx/Rx), as well as a sweep jamming algorithm operating within the 2.5 GHz frequency band of a Wireless Fidelity channel, utilizing a 20 MHz bandwidth. The jamming power is set at 30 dB, and the Tx signal power ranges from 20 to 50 dB. The demodulation threshold is 8 dB, and the carrier signal frequency is 5GHz, with a data rate of 2 Mbps and BPSK modulation. A channel switching cost of 0.2 is considered, along with a discount factor of 0.96 and a minibatch size of 64. Fig. 2 illustrates the scenario of environment rendering with a sweep jammer following different channels over different time intervals.

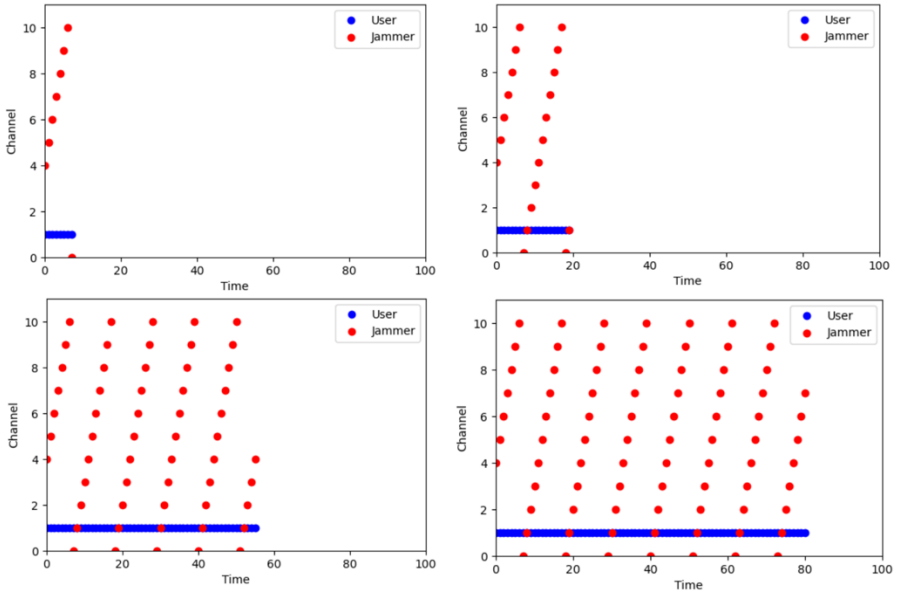

Fig. 2. Rendering of environment with user (transmitter) and sweep jammer

4.1. Implementation of sweep jamming attack

In sweep jamming, the jammer emits signals that sweep across a range of frequencies, disrupting any communication that uses those frequencies. The mathematical modeling for sweep jamming can be shown as follows:

Let the jamming signal be denoted by $x(t)$, and the received signal be denoted by $y(t)$. Assume that the received signal $y(t)$ is the sum of the transmitted signal $x(t)$ and $n(t)$, given as follows:

$$y(t) = x(t) + n(t). \tag{15}$$

The jammer's signal $x(t)$ is swept across a range of frequencies, which can be represented as a function of time *f*(*t*). The frequency sweep can be modeled as a linear chirp signal given by

$$f(t) = f_0 + kt, \tag{16}$$

where f_0 is the starting frequency, k is the sweep rate, and t is time. The jamming signal $x(t)$ can be obtained by modulating a carrier signal with the frequency sweep $f(t)$ using BPSK modulation. Mathematically, this can be represented as

(17) $$x(t) = A\cos(2\pi f(t)t + \varphi),$$

where A denotes amplitude, and φ denotes the phase of the carrier signal. An example of sweep jamming can be demonstrated as follows: Assume that a communication system uses a frequency band of 2.5 GHz. The jammer sweeps across this frequency band with a sweep rate of 20 MHz/μs, starting from 2.5 GHz. The jammer's transmitted signal $x(t)$ can be obtained by modulating a carrier signal with the frequency sweep using BPSK modulation. Any communication using frequencies within the 2.5 GHz band is disrupted by the transmission of the jamming signal.

4.2. Implementation of smart jamming attack

The proposed study has adopted the same mechanism, i.e., DDPG which is employed in the implementation of the proposed anti-jamming scheme. However, the difference is the basis of the reward function. The agent, designed to jam the signal, aims to maximize its reward by disrupting the signal. The mathematical equation used to determine the Reward factor is as follows:

(18) $$R_{t2} = -1 \times R_{t1}.$$

The agent responsible for the jamming process receives a positive reward, whereas the agent responsible for the anti-jamming process receives a negative reward. In this proposed study, two agents are used – an intelligent jammer and an intelligent anti-jammer. The agent assigned to the smart jamming task seeks to maximize its reward by disrupting the signal, while the agent assigned to the anti-jamming process aims to minimize the reward value for the jamming agent by developing a new action policy to counter the signal disruption. The reward function for the jamming agent is positive, while the reward function for the anti-jamming agent is negative. This creates a non-cooperative interaction between the two agents, where each agent tries to maximize its own reward at the expense of the other agent.

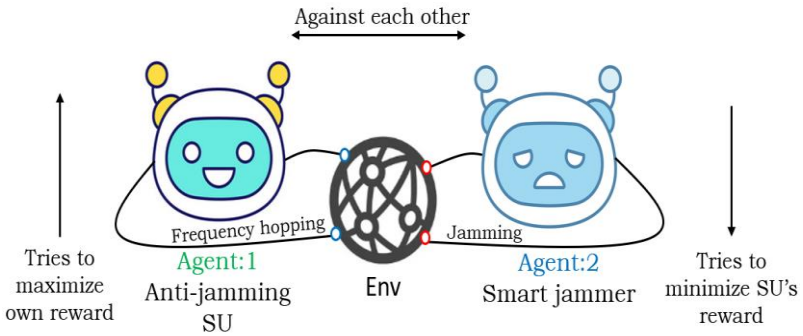

Fig. 3. Adversarial learning scenario

Fig. 3 depicts an example scenario with two agents in a game, Agent-1 and Agent-2. Agent-1 wants to score a goal, while Agent-2 wants to stop Agent-1 from scoring. In this case, Agent-1's reward function will be positive if he or she score a goal, while Agent-2's reward function will be negative if Agent-1 scores a goal. This creates a competitive environment where each agent tries to outsmart the other to achieve his or her individual goals. In the proposed study, the jamming agent and

anti-jamming agent both act rationally to maximize their respective rewards based on their actions, resulting in an adversarial learning scenario. Following are the real-time operational flow process:

- Sensing the Environment. As the cognitive radio operates, it continually monitors its RF environment. This real-time data forms the state input for our DDPG agent.
- Action Decision. The integrated DDPG agent processes the state information and predicts the best action based on its training. Such actions could include switching channels, modulating transmission power, or any other anti-jamming strategy.
- Action Execution. The proposed action is then executed in real-time. For instance, if the agent decides to switch channels, the software-defined components of the radio make the switch immediately.

4.3. Performance analysis

Performance evaluation is done with respect to power factor and SNR over progressive time slots. The study also has conducted a comparative analysis considering other variants of reinforcement learning algorithms.

The time-frequency analysis for sweep jamming and transmitter inside the suggested simulation environment is shown in Fig. 4. The channel that the jammer is aiming to block is depicted in blue in the illustration, while data transmission is shown in orange. The red signal in the diagram is the jammed signal, and it implies that any data packets sent across the jammed channel will suffer a reduction in SNR, which will result in transmission errors.

The time-frequency analysis in Fig. 5 shows how both transmitters and intelligent jammers strive to maximize the use of sub-band channels for the accomplishment of their individual goals. The transmitter must be aware of potential sub-channels in order to prevent jamming, while the jammers continuously predict their sub-band channels. The received signal's SNR will drop if a user tries to send data packets over the jammed channel, which will result in transmission errors.

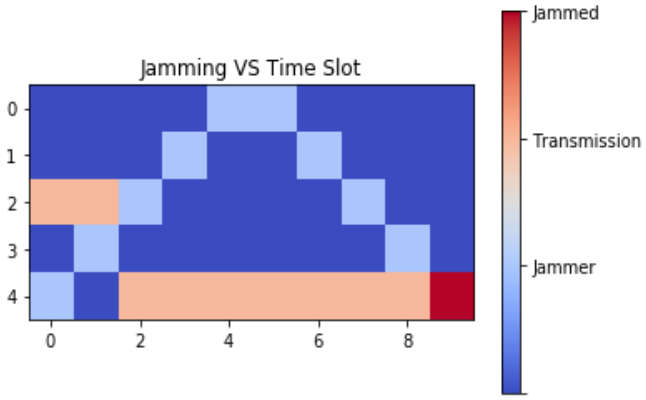

Fig. 4. Time-frequency analysis of sweep jammer and transmission

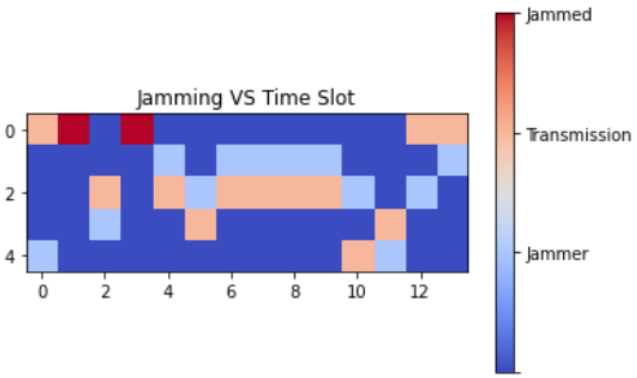

Fig. 5. Time-frequency analysis of smart jammer and transmission

Table 3. Quantified values for power analysis

Time steps	Q-learning	DQN	DDPG	Adversarial learning (proposed)
1	1.97	1.03	1.01	1.01
10	0.50	0.41	0.37	0.14
20	0.14	0.21	0.14	0.02
30	0.21	0.11	0.06	0.01
40	0.79	0.05	0.03	0.00
50	0.30	0.03	0.02	0.00
60	0.28	0.06	0.00	0.00
70	0.19	0.09	0.01	0.01
80	0.22	0.10	0.01	0.01
90	0.01	0.08	0.00	0.00
100	0.11	0.00	0.00	0.00

Table 3 presents the quantified outcome of power consumption for the performance analysis of different agent learning techniques such as Q-learning, DQN, DDPG, and proposed agent mechanism trained dynamically (i.e., trained against sweep jamming and smart jamming as adversarial learning mechanisms). The graphical representation of the proposed scheme is illustrated in Fig. 6.

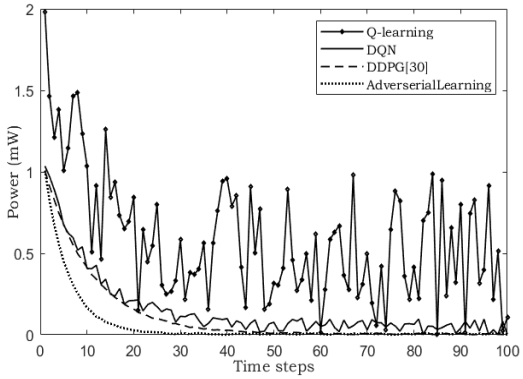

Fig. 6. Analysis of the power cost

As shown in Fig. 6, the DQN and DDPG demonstrate minor differences in their performance, with DQN being more effective than Q-learning due to its utilization of neural networks to maximize action value for a specific task. However, DDPG is considered more reliable in terms of performance and is better suited to continuous action spaces due to the use of actor and critic neural networks.
In the proposed adversarial learning method, two DDPG algorithms, one for smart jamming and the other for anti-jamming are trained to learn from one another and create better rules. The algorithm's instability and energy consumption are decreased as a result of this strategy.

Table 4. Quantified values for SNR analysis

Time steps	Q-learning	DQN	DDPG	Adversarial learning (proposed)
1	3.17	3.76	3.7	4.48
10	4.63	6.02	6.01	6.74
20	5.6	6.71	6.78	7.4
30	6.0	7.18	7.15	7.8
40	6.19	7.42	7.4	8.0
50	6.61	7.6	7.65	8.3
60	6.57	7.8	7.83	8.4
70	6.89	8.02	7.98	8.6
80	6.39	8.0	8.09	8.81
90	6.59	8.2	8.28	8.93
100	7.02	8.3	8.35	9.05

Table 4 presents the quantified outcome of power consumption for the performance analysis of different agent learning and its graphical outcome has been demonstrated in Fig. 7. The suggested algorithm's capacity to continually enhance performance through a competitive learning process between smart jammer and anti-jamming agents makes it special. The suggested algorithm also adjusts to dynamic jamming circumstances and modifies policies as necessary. Table 5 indicates that the proposed system exhibits greater improvement in reducing power consumption, and enhancing signal quality.

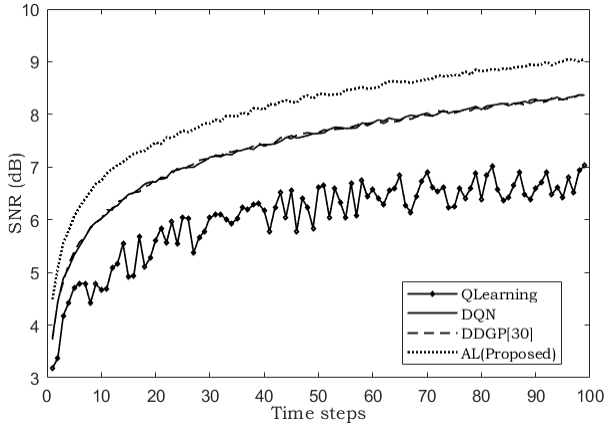

Fig. 7. Analysis of the SNR

Table 5. Comparison in outcome

Anti-jamming algorithm	Improvement in power	Improvement in SNR	Processing time
Q-learning	35.81%	81.59%	0.98791 s
DQN	76.29%	82.63%	0.59851 s
DDPG	87.57%	98.41%	0.26619 s
Proposed	~88%	~90%	0.3645 s

The proposed study also considers the analysis of processing time complexity as an important parameter, which shows the computational efficiency, and action response speed, which is highly desirable in real-world context. The time and space complexity of each model are discussed as follows:

The time and space complexity for Q-learning can be described $O(|\text{States}| \times |\text{Action}|)$ for each episode. This complexity arises because Q-learning must update its Q-table for every state-action combination. In our experiments, Q-learning exhibited a processing time of 0.98791 s. The main computational cost in DQN arises from forward and backward passes through the neural network. For a neural network with L layers, each having N neurons and the weights of the neural network, hence the complexity can be described as $O(L \times N^2)$. The experimental analysis shows DQN demonstrated a processing time of 0.59851 s. Like DQN, DDPG uses neural networks for both the actor and the critic. If both networks have L layers with N neurons the time complexity is $O(L \times N^2)$. DDPG achieved a processing time of 0.26619 s in our experiments. This is indicative of the efficiency of the model after it has been trained. The training involves two DDPG agents (anti-jamming and jammer). Therefore, the time complexity is roughly double that of a single DDPG, i.e., $O(2 \times L \times N^2)$. Our proposed Adversarial Learning approach recorded a processing time of 0.3645 s. Despite the involvement of two DDPG agents during training, the processing time remains efficient. This efficiency in real-world scenarios stems from the model's comprehensive training against dynamic jamming strategies, allowing the anti-jamming agent to formulate a highly responsive action policy.

5. Conclusion

A model-free, off-policy agent based self-exploration mechanism is presented as an effective and clever anti-jamming strategy in the proposed system. The proposed solution not only eliminates the unnecessary overhead associated with continuous action space but also demonstrates greater intelligence than previous efforts and has a significant positive impact on the environment. The simulation results demonstrate that our proposed reinforcement learning-based anti-jamming solution can effectively defend against sweep and smart jamming attacks. The DDPG agent learns to choose the optimal transmission power to minimize the impact of the jamming signal and achieves a high SNR even in the presence of jamming signals. Our proposed method proves to be robust against rule-based and intelligent jammers and offers optimal power consumption and improved SNR. Based on simulation results, the proposed anti-jamming approach has been demonstrated to be more effective than popular approaches such as Q-learning and DQN.

References

1. L i, S., et al. 5G Internet of Things: A Survey. – Journal of Industrial Information Integration, Vol. **10**, 2018, pp. 1-9.
2. K i n z a, B. A., et al. Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios. – IEEE Access, Vol. **8**, 2020, pp. 23022-23040.
3. S i d d i k o v, I., et al. CRN and 5G Based IoT: Applications, Challenges and Opportunities. – In: International Conference on Information Science and Communications Technologies (ICISCT'21), IEEE, 2021, pp. 1-5.
4. Z i k r i a, Y. B., et al. Internet of Things (IoT) Operating Systems Management: Opportunities, Challenges, and Solution. – Sensors (Basel, Switzerland), Vol. **19**, 2019, No 8, p. 1793.
5. S h a k h a k a r m i, N. 5G Wireless Communications Systems: Heterogeneous Network Architecture and Design for Small Cells, D2D Communications (Low Range, Multi-Hop) and Wearable Healthcare System on Chip (ECG, EEG) for 5G Wireless. – Int. J. Comput. Sci. Issues, Vol. **13**, 2016, No 6, pp. 34-45.
6. R a t h e e, G., et al. CRT-BIoV: A Cognitive Radio Technique for Blockchain-Enabled Internet of Vehicles. – IEEE Transactions on Intelligent Transportation Systems: A Publication of the IEEE Intelligent Transportation Systems Council, Vol. **22**, 2021, No 7, pp. 4005-4015.
7. L i u, M., et al. DSF-NOMA: UAV-Assisted Emergency Communication Technology in a Heterogeneous Internet of Things. – IEEE Internet of Things Journal, Vol. **6**, 2019, No 3, pp. 5508-5519.
8. A t t a r, A., et al. A Survey of Security Challenges in Cognitive Radio Networks: Solutions and Future Research Directions. – Proceedings of the IEEE. Institute of Electrical and Electronics Engineers, Vol. **100**, 2012, No 12, pp. 3172-3186.
9. B r i t t, C. L., D. F. P a l m e r. Effects of CW Interference on Narrow-Band Second-Order Phase-Lock Loops. – IEEE Transactions on Aerospace and Electronic Systems, Vol. **AES-3**, 1967, No 1, pp. 123-135.
10. P i r a y e s h, H., H. Z e n g. Jamming Attacks and Anti-Jamming Strategies in Wireless Networks: A Comprehensive Survey. – IEEE Communications Surveys & Tutorials, Vol. **24**, 2022, No 2, pp. 767-809. DOI:10.1109/comst.2022.3159185.
11. A r e f, M. A., et al. Survey on Cognitive Anti-Jamming Communications. – IET Communications, Vol. **14**, 2020, No 18, pp. 3110-3127.
12. K h a n, A. A., et al. Cognitive Radio for Smart Grids: Survey of Architectures, Spectrum Sensing Mechanisms, and Networking Protocols. – IEEE Communications Surveys & Tutorials, Vol. **18**, 2016, No 1, pp. 860-898.
13. E l M r a b e t, Z., et al. Cyber-Security in Smart Grid: Survey and Challenges. – Computers & Electrical Engineering, Vol. **67**, 2018, pp. 469-482.
14. R a h m a n i l, M. Frequency Hopping in Cognitive Radio Networks: A Survey. – In: Proc. of IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE'2015), IEEE, 2015, pp. 1-6.
15. S h i v a m, J., et al. A Detail Survey of Channel Access Method for Cognitive Radio Network (CRN) Applications toward 4G. – South Asian Research Journal of Engineering and Technology, Vol. **3**, 2021, No 1, pp. 31-41.
16. T h a k u r, P., G. S i n g h. Power Management for Spectrum Sharing in Cognitive Radio Communication System: A Comprehensive Survey. – Journal of Electromagnetic Waves and Applications, Vol. **34**, 2020, No 4, pp. 407-461.
17. S h a r m a, R. K., D. B. R a w a t. Advances on Security Threats and Countermeasures for Cognitive Radio Networks: A Survey. – IEEE Communications Surveys & Tutorials, Vol. **17**, 2015, No 2, pp. 1023-1043.
18. H a n, C., et al. Intelligent Anti-Jamming Communication Based on the Modified Q-learning. – Procedia Computer Science, Vol. **131**, 2018, pp. 1023-1031.
19. U p r e t y, A., D. B. R a w a t. Reinforcement Learning for IoT Security: A Comprehensive Survey. – IEEE Internet of Things Journal, Vol. **8**, 2021, No 11.

20. S i n g h, S., A. T r i v e d i. Anti-Jamming in Cognitive Radio Networks Using Reinforcement Learning Algorithms. – In: Proc. of 9th International Conference on Wireless and Optical Communications Networks (WOCN'12), IEEE, 2012.
21. N a l l a r a s a n, V., K. K o t t u r s a m y. Cognitive Radio Jamming Attack Detection Using an Autoencoder for CRIoT Network. – In: Wireless Personal Communications. 2021, pp. 1-17.
22. I b r a h i m, K., et al. Anti-Jamming Game to Combat Intelligent Jamming for Cognitive Radio Networks. – IEEE Access: Practical Innovations, Open Solutions, Vol. **9**, 2021, pp. 137941-137956.
23. X i a o, L., et al. Power Control with Reinforcement Learning in Cooperative Cognitive Radio Networks against Jamming. – The Journal of Supercomputing, Vol. **71**, 2015, No 9, pp. 3237-3257.
24. S u d h a, Y., V. S a r a s v a t h i. An Intelligent Anti-Jamming Mechanism against Rule-Based Jammer in Cognitive Radio Network. – International Journal of Advanced Computer Science and Applications (IJACSA), Vol. **13**, 2022, No 3.
25. S u d h a, Y., V. S a r a s v a t h i. A Model-Free Cognitive Anti-Jamming Strategy Using Adversarial Learning Algorithm. – Cybernetics and Information Technologies, Vol. **22**, 2022, No 4, pp. 56-72.
26. T h i e n, H. T., et al. A Transfer Games Actor-Critic Learning Framework for Anti-Jamming in Multi-Channel Cognitive Radio Networks. – IEEE Access: Practical Innovations, Open Solutions, Vol. **9**, 2021, pp. 47887-47900.
27. H u a n g, J.-F., et al. Anti-Jamming Rendezvous Scheme for Cognitive Radio Networks. – IEEE Transactions on Mobile Computing, Vol. **16**, 2017, No 3, pp. 648-661.
28. H a n a w a l, M. K., et al. Joint Adaptation of Frequency Hopping and Transmission Rate for Anti-Jamming Wireless Systems. – IEEE Transactions on Mobile Computing, Vol. **15**, 2016, No 9, pp. 2247-2259.
29. G a o, Y., et al. Game Theory-Based Anti-Jamming Strategies for Frequency Hopping Wireless Communications. – IEEE Transactions on Wireless Communications, Vol. **17**, 2018, No 8, pp. 5314-5326.
30. Z h a n g, X., et al. Joint Channel and Power Optimisation for Multi-user Anti-jamming Communications: A Dual Mode Q-learning Approach. – IET Communications, Vol. **16**, 2022, No 6, pp. 619-633.
31. C a d e a u, W., et al. Markov Model Based Jamming and Anti-Jamming Performance Analysis for Cognitive Radio Networks. – Communications and Network, Vol. **6**, 2014, No 2, pp. 76-85.

Received: 10.04.2023; Second Version: 09.08.2023; Third Verion: 26.08.2023; Accepted: 14.09.2023