Scientific African 27 (2025) e02523

Contents lists available at ScienceDirect

Scientific African

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

Deep reinforcement learning agents for dynamic spectrum access in television whitespace cognitive radio networks☆

Udeme C. Ukpong [a,b,*], Olabode Idowu-Bismark [a,b], Emmanuel Adetiba [a,b,e], Jules R. Kala [c], Emmanuel Owolabi [d], Oluwadamilola Oshin [a,b], Abdultaofeek Abayomi [f,g], Oluwatobi E. Dare [a,b]

[a] *Department of Electrical and Information Engineering, Covenant University, Ota, Nigeria*
[b] *Covenant Applied Informatics & Communication African Center of Excellence (CApIC-ACE) Covenant University, Ota, Nigeria*
[c] *International University of Grand-Bassam, Grand-Bassam, Côte d'Ivoire*
[d] *University of Pretoria, Pretoria, South Africa*
[e] *HRA, Institute for Systems Science, Durban University of Technology, Durban, South Africa*
[f] *HRA, Walter Sisulu University, East London 5200, South Africa*
[g] *Innovation and Advanced Science Research Group (IASRG), Summit University, PMB 4412, Offa, Kwara, Nigeria*

ARTICLE INFO

Editor: DR B Gyampoh

Keywords:
Cognitive radio networks
Deep reinforcement learning
DQN
Dynamic spectrum access
QR-DQN
Television whitespace;RFRL gym

ABSTRACT

Businesses, security agencies, institutions, and individuals depend on wireless communication to run their day-to-day activities successfully. The ever-increasing demand for wireless communication services, coupled with the scarcity of available radio frequency spectrum, necessitates innovative approaches to spectrum management. Cognitive Radio (CR) technology has emerged as a pivotal solution, enabling dynamic spectrum sharing among secondary users while respecting the rights of primary users. However, the basic setup of CR technology is insufficient to manage spectrum congestion, as it lacks the ability to predict future spectrum holes, leading to interferences. With predictive intelligence and Dynamic Spectrum Access (DSA), a CR can anticipate when and where other users will be using the radio frequency spectrum, allowing it to overcome this limitation. Reinforcement Learning (RL) in CRs helps predict spectral changes and identify optimal transmission frequencies. This work presents the development of Deep RL (DRL) models for enhanced DSA in TV Whitespace (TVWS) cognitive radio networks using Deep Q-Networks (DQN) and Quantile-Regression (QR-DQN) algorithms. The implementation was done in the Radio Frequency Reinforcement Learning (RFRL) Gym, a training environment of the RF spectrum designed to provide comprehensive functionality. Evaluations show that the DQN model achieves a 96.34 % interference avoidance rate compared to 95.97 % of QRDQN. Average latency was estimated at 1 millisecond and 3.33 milliseconds per packet, respectively. Therefore DRL proves to be a more flexible, scalable, and adaptive approach to dynamic spectrum access, making it particularly effective in the complex and constantly evolving wireless spectrum environment.

☆ This work was supported by Covenant Applied Informatics and Communication Africa Centre of Excellence (CApIC-ACE), Covenant University with part funding to the first author(UU) through the Google Award for TensorFlow Outreaches in Colleges awarded to the third author (EA).

* Corresponding author.

E-mail addresses: udeme.ukpongpgs@stu.cu.edu.ng (U.C. Ukpong), olabode.idowu-bismark@covenantuniversity.edu.ng (O. Idowu-Bismark), emmanuel.adetiba@covenantuniversity.edu.ng (E. Adetiba), kala.j@iugb.edu.ci (J.R. Kala), owolabi.emmanuel@gmail.com (E. Owolabi), damilola.adu@covenantuniversity.edu.ng (O. Oshin), taofeekab@yahoo.com (A. Abayomi), oluwatobi.darepgs@stu.cu.edu.ng (O.E. Dare).

https://doi.org/10.1016/j.sciaf.2024.e02523
Received 31 October 2024; Received in revised form 9 December 2024; Accepted 24 December 2024
Available online 26 December 2024

Introduction

Access to radio frequencies is crucial for many vital industries and military sectors, such as television and radio transmission, mobile communications, transportation, satellites and IoT (Internet of Things) [1]. As the global number of wireless devices and services continues to increase, demand for wireless resources is growing exponentially. With this exponential increase of both internet users and connected devices, spectrum saturation has necessitated more efficient spectrum management strategies such as utilising TV white spaces (TVWS), which are the underutilised bands within the TV broadcast spectrum [2,3]. Cognitive Radio Networks (CRNs) aim to dynamically access these TVWS without interfering with incumbent users [4], but traditional methods often fall short in managing the dynamic and diverse requirements of TVWS [5].

Reinforcement Learning (RL) offers predictive capabilities that enable signals to preemptively adapt to the movements of other signals in the spectrum and changing channel conditions, thereby identifying the optimal transmission frequency [6]. Extensive research has been dedicated to exploring RL-based protocols for spectrum assignment and enhancing wireless communication. These efforts have demonstrated that RL can significantly reduce congestion in the spectrum, as seen in works done by [7,8] and [9].

In today’s increasingly complex network environments, optimising network performance has become a real challenge [10]. A combination of deep learning and reinforcement learning, known as deep reinforcement learning, shows great promise for improving the intelligence of wireless communication systems and meeting this challenge [11,12].

Deep reinforcement learning represents a cutting-edge machine learning paradigm, particularly well-suited for addressing the challenges posed by Dynamic Spectrum Access (DSA) [13]. DeepRL distinguishes itself by harnessing the power of deep neural networks to approximate intricate functions, a capability that proves instrumental in navigating the complexities associated with high-dimensional state spaces inherent to DSA scenarios [14,15].

In the context of DSA, where the allocation and utilization of radio frequency spectrum dynamically evolves, DeepRL excels by automatically learning hierarchical features directly from raw input data [16]. This stands in stark contrast to traditional approaches that necessitate manual feature engineering. The ability of DeepRL to autonomously identify and extract relevant features from raw observations streamlines the modeling process [17], allowing for more adaptive and efficient decision-making in response to the dynamic nature of spectrum access scenarios [18].

Given the large state space and partially observed nature of spectral management among wireless connected devices [19], incorporating DeepRL methods into the design of dynamic spectrum access algorithms holds significant potential for delivering effective solutions to the complex spectrum access challenges in the TV whitespace radio frequency band [20]. This potential serves as the primary motivation for this research. This work involves simulating a cognitive radio network to evaluate reinforcement learning algorithms within the wireless communication Radio Frequency spectrum environment. Two models are developed using Deep Reinforcement Learning techniques—DQN and QRDQN—for Dynamic Spectrum Access in Cognitive Radio Networks. A comparative analysis of these two advanced algorithms is conducted to offer insights into their real-time performance in dynamic spectrum access scenarios, enhancing our understanding of their operational efficiency and reliability in real-world applications.

Review of related works

Various Machine Learning (ML) paradigms have been applied to solve problems in dynamic spectrum access; however, each learning paradigm comes with its strengths and accompanying weaknesses. Supervised and unsupervised learning face notable challenges in addressing the dynamic and real-time nature of DSA. Supervised learning relies on labeled datasets, which are difficult to maintain in rapidly changing spectrum environments, leading to outdated models and reduced adaptability. Unsupervised learning, while independent of labeled data, struggles with accurately capturing the complex and evolving patterns of spectrum usage due to the dynamic presence of primary and secondary users, interference considerations, and varying demands. These limitations highlight the need for more adaptable approaches, such as reinforcement learning and deep reinforcement learning, which are better equipped to handle the intricate and dynamic characteristics of DSA scenarios [21].

In [11], Luong et al. presented a comprehensive survey and tutorial on the practical uses of deep reinforcement learning in communications and networking. They identified and categorized important applications of DeepRL in areas such as network access, adaptive data rate regulation, wireless caching, data offloading, network security, connection preservation, traffic routing, and data gathering. The survey covered various DeepRL approaches like Asynchronous Advantage Actor-Critic (A3C), Deep Q-Network (DQN), Double DQN (DDQN), Deep Recurrent Q-Learning (DRQN), and Dynamic Adaptive Streaming over HTTP (DASH), highlighting DeepRL’s potential to solve complex networking problems. They also provided insights into the distribution of research efforts in this field and discussed significant obstacles, unresolved issues, and future research directions.

Naparstek & Cohen proposed a deep multi-user reinforcement learning method for distributed dynamic spectrum access in multichannel wireless networks [15]. Their goal was to optimize channel access without requiring real-time coordination or communication between users. The proposed technique enabled decentralized learning through the analysis of acknowledgment (ACK) signals, which led to increased channel throughput by reducing idle time slots and collisions. Comprehensive numerical tests demonstrated the adaptability and efficiency of the method, achieving a channel throughput twice as high as the slotted-Aloha with optimal transmission probability.

A novel approach using a Deep Q-Network (DQN) to learn an optimal policy for multichannel access in a partially observable environment was introduced by Wang et al. in [22]. They formulated the dynamic multichannel access problem as a partially observable Markov decision process (POMDP) to account for partial observations and potential channel correlations. The authors compared the DQN framework with Myopic and Whittle Index-based heuristics, demonstrating that the DQN approach achieved near-optimal performance in complex scenarios. The effectiveness of the DQN framework was validated using both synthetic and testbed-based datasets.

In [23], Liu et al. introduced a Deep Reinforcement Learning-Based Dynamic Channel Allocation (DRL-DCA) technique for multibeam satellite systems. The algorithm aimed to optimally assign channels to user terminals (UTs) to maximize system throughput and minimize co-channel interference. The DRL-DCA algorithm combined deep learning with reinforcement learning, employing a deep convolutional neural network (CNN) to extract valuable features from the system state. Simulation results showed that the DRL-DCA algorithm outperformed traditional methods like Fixed Channel Assignment (FCA) and Interference Mitigation Dynamic Channel Allocation (IM-DCA), enhancing carried traffic by 24.4 % to 41.7 % and spectrum efficiency by 21.7 %.

Yu et al. in [24] proposed the Deep-reinforcement Learning Multiple Access (DLMA) protocol for heterogeneous wireless networks. The DLMA protocol utilized deep reinforcement learning to optimize channel access methods in a diverse networking environment with multiple MAC protocols. The authors developed a multi-dimensional reinforcement learning framework to maximize global α-fairness in DLMA. Simulations demonstrated that the DLMA protocol achieved near-optimal total data transfer rates and fairness, showing resilience to suboptimal parameter configurations and quicker convergence compared to standard reinforcement learning methods. The DLMA protocol effectively optimized channel access strategies and achieved defined global objectives in diverse wireless network environments.

This work differentiates itself from existing approaches in the domain by integrating advanced Deep Reinforcement Learning (DRL) algorithms, specifically DQN and QRDQN, to optimize channel selection for secondary users in a cognitive radio network environment. Unlike traditional methods, this approach focuses on dynamic, data-driven decision-making with a more nuanced exploration-exploitation tradeoff, resulting in enhanced spectrum efficiency and adaptability. Additionally, the integration of a geolocation database for spectrum sensing combined with a structured performance evaluation framework provides a more robust and scalable solution compared to previous works.

Methodology

System model

Fig. 1 presents the system architecture, which comprises four primary components. The first component represents the cognitive radio network environment. The second component is the spectrum sensing block, which incorporates a geolocation database. The third block represents the DeepRL agents, while the final block is dedicated to performance evaluation. Detailed explanations of each component are provided in the following sections to offer a comprehensive understanding of the proposed framework.

Design and simulation of cognitive radio network environment

For the design and simulation of a cognitive radio network environment, Radio Frequency Reinforcement Learning (RFRL) Gym, a Reinforcement Learning Testbed specifically tailored for Cognitive Radio Applications [25], based on OpenAI Gym is employed. This testbed provides a versatile platform for creating and evaluating various scenarios in cognitive radio networks. It allows for the definition of network topologies, mobility patterns, and communication protocols, facilitating the emulation of dynamic spectrum environments. As shown in Fig. 2, a network scenario is designed with 2 primary users, 20 secondary users and 10 available channels. The 20 secondary users aim to operate concurrently within the coverage area of the primary users. This concurrent operation can result in interference, potentially degrading the quality of service for the primary users.

This network configuration is representative of a typical real-world network environment and can be effectively simulated using the computational resources available and used in this work (Table I).

Spectrum holes

To implement the spectrum sensing method, the project utilizes geolocation database. Specifically, the Reference Geolocation Spectrum Database from Independent Communications Authority of South Africa is used [26]. This database contains information about the occupancy of frequency bands in specific geographic locations. Information from this database is downloaded into the RFRL Gym to determine available channels. By leveraging this database, the cognitive radio nodes identify spectrum holes in the TV band, enabling efficient access to available frequencies.

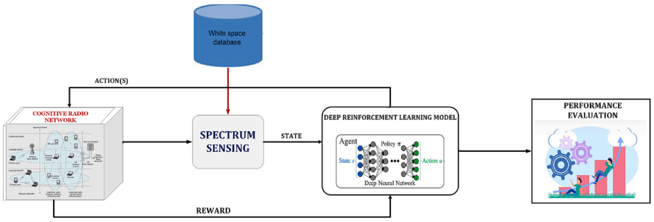


Fig. 1. Overview of system architecture. Illustrates the four primary components: the cognitive radio network environment, spectrum sensing with a geolocation database, DeepRL agents, and the performance evaluation block.

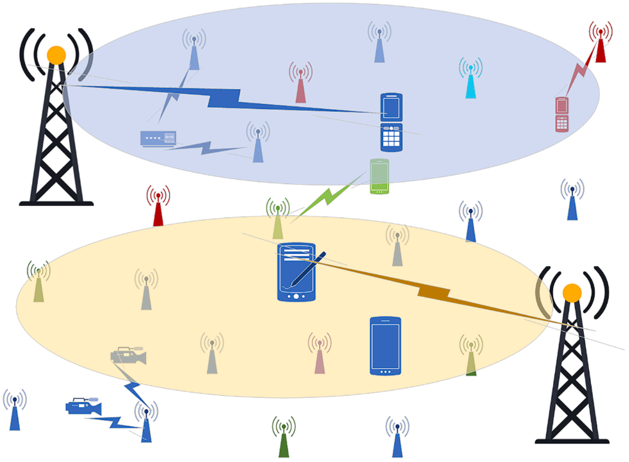


Fig. 2. Cognitive Radio Network scenario. Depicts a network scenario with 2 primary users, 20 secondary users, and 10 available channels. Secondary users operate concurrently within the primary users' coverage area, potentially causing interference and degrading the primary users' quality of service.

Table I
Virtual machine hardware specification.

Component	Specification
Form Factor	2 U Rack
Processors	Intel Xeon Scalable processors
Memory	2 TB (24 DIMM slots)
Storage	Up to 16 × 2.5″ or 8 × 3.5″ drives (HDD/SSD)
RAID Controller	PERC H730P, H740P, HBA330
Network Interface	4 × 1GbE LOM; optional 10GbE or 25GbE network cards
Expansion Slots	8 x PCIe Gen3 slots
Power Supply	2 × 1100 W, 1600 W, or 2000 W hot-plug redundant PSU
Management	iDRAC9 with Lifecycle Controller
GPU Support	Up to 3 × 300 W or 6 × 150 W GPUs
Dimensions	Height: 8.68 cm (3.41 in) Width: 48.24 cm (18.99 in) Depth: 71.09 cm (27.99 in)
Weight	Maximum: 28.3 kg (62.5 lbs)

Model development

Stable-Baselines3 - a Reliable Reinforcement Learning framework for Implementations of Reinforcement Learning algorithms is employed. Two models are trained using the Deep Q Network (DQN) and Quantile Regression DQN (QR-DQN) algorithms.

i) DQN algorithm

DQN is a model-free, off-policy reinforcement learning algorithm that combines Q-learning with deep neural networks. The goal is to learn the action-value function *Q(s,a),* which estimates the expected return (sum of future rewards) of taking action *a* in state *s* and following the optimal policy thereafter [27].

$$Q_{(s,a)} = (1-\alpha)Q_{(s,a)+} + \alpha\left(r + \gamma maxa' Q_{(s',a')}\right) \tag{1}$$

In (1), the Q Function *Q(s, a)* is being referred to twice. Firstly, (1 - α)$Q_{(s,\ a)}$ to retrieve the present state-action value so as to update its value, and secondly, to get the ***"target"*** value for the subsequent Q value for the next state-action (i.e., Q as in: $r + \gamma \max_{a'} Q_{(s',\ a')}$. Instead of learning from consecutive samples, DQN uses a replay buffer to store experiences (s,a,r,s′) and samples mini-batches uniformly to break correlations between consecutive samples and improve stability. This is generally referred to as 'experience replay.' To stabilize training, DQN uses the separate target network $Q_{\theta'}$ with parameters θ' that are copied from the main network Q_θ every few steps. The Q-learning update rule is given by (2) below:

$$y = r + \gamma \underset{a'}{max} Q_{\theta'}(s', a') \tag{2}$$

where *r* is the reward, γ is the discount factor, and *s′* is the next state. Eq. (3) shows the loss function for updating the parameters θ of the Q-network.

$$L(\theta) = \mathrm{E}_{(\mathrm{s,a,r,s'})} \sim D\left[(y - Q_\theta(s,a))^2\right] \tag{3}$$

where D is the replay buffer from which experiences are sampled.

ii) Quantile Regression Deep Q-Network (QR-DQN)

QR-DQN extends DQN by learning the distribution of returns instead of just the expected return. It does this by approximating the quantiles of the return distribution, which allows for better risk-sensitive decision making. Instead of estimating a single Q-value, QR-DQN estimates multiple quantiles of the return distribution for each state-action pair. This algorithm also employs the Quantile Huber loss function which combines the Huber loss with the quantile regression loss, providing robustness to outliers while focusing on specific quantiles of the return distribution.

QR-DQN estimates the quantile function θ_i for $i = 1,\ldots,$N where N is the number of quantiles. The target for quantile i is captured in (4) below:

$$y = r + \gamma\theta_i'(s', a') \tag{4}$$

where θ'_i are the quantiles of the target network. The quantile Huber loss for updating the network parameters is:

$$L(\theta) = E_{(s,a,r,s')} \sim D\left[\frac{1}{N}\sum_{N}^{i=1}\sum_{N}^{j=1}\rho k\left(y_i - \theta_j(s,a)\right)\right] \tag{5}$$

where ρ_k is the Huber loss defined as:

$$\begin{cases} \frac{1}{2}u^2 \text{ if } |\mathrm{u}| \le k \\ k\left(|u| - \frac{1}{2}k\right) \textit{ otherwise} \end{cases} \tag{6}$$

The variable k represents the residuals, which are the differences between the values actually obtained and predicted values. The function exhibits quadratic behaviour for small values of u and linear behaviour for big values, with the same values and slopes when |u|=k.

Both algorithms represent significant advancements in the field of deep reinforcement learning. The DeepRL models are designed to make intelligent decisions for Dynamic Spectrum Access based on the information obtained from spectrum sensing. Scenario data including information on channel quality, interference, and available spectrum, are used for training to ensure the models learn effective spectrum-sharing policies. The results of both models are evaluated to compare the performance of both algorithms.

iii) Model Training Parameters

Table II outlines the training parameters used in training the DQN and QR-DQN agents. This includes 1000 epochs, 100 episodes per epoch, 10 communication channels, 2 primary (constant) users, and 20 secondary users. These parameters define the structure and dynamics of the training environment, specifying the number of iterations, interactions, and resource allocations involved in the simulation. The reward currently implemented in the RFRL Gym environment for the RL agent is the DSA reward. DSA mode can be used to simulate a situation where an RL agent wants to avoid interference, either by avoiding a channel with bad channel conditions or a channel with other users and/or adversaries. The DSA reward values are shown in (7).

$$\text{DSA reward} = \begin{cases} 1, \ \textit{no collision} \\ -1, \ \textit{collision} \end{cases} \tag{7}$$

The discount factor is applied to modify the reward based on the agent's action of staying in the same channel or switching channels. Specifically:

- If the agent remains on the same channel, the reward is increased by 0.5, incentivizing stable behavior.
- If the agent switches to a different channel, the reward is reduced by 0.5, discouraging unnecessary movement.

Table II
Model training parameters.

Parameter	Value
No. of epochs	1000
No. of episodes	100
No. of channels	10
No. of primary (constant users)	2
No. of secondary users	20

This adjustment acts as a form of regularization, ensuring that the agent is able to prioritize stability in channel selection unless a move is necessary.

Results and discussion

Performance evaluation

The evaluation of the DeepRL model's performance was conducted using multiple key performance indicators (KPIs):

i) **Mean Episode Reward:** This is the average reward obtained by the agent per episode over a specified number of episodes. It gives a direct measure of the agent's performance. Higher mean episode rewards typically indicate better performance in achieving the desired task**.** If we denote *Ri* as the total reward received in the *i th* episode, and we have N episodes, the Mean Episode Reward $\bar{R}$ is calculated as shown in (8):

$$\overline{R} = \frac{1}{N}\sum\frac{N}{i=1}R_i \tag{8}$$

Where N is the total number of episodes, and $\mathrm{R_i}$ is the total reward for the *i* th episode

ii) **ii) Exploration Rate (ϵ):** This is the probability of choosing a random action (exploration) rather than the action suggested by the current policy (exploitation). It balances exploration of new actions and exploitation of known rewarding actions. A high exploration rate means the agent is trying out more new actions, while a low rate means the agent is relying more on the learned policy. To ensure effective exploration, ϵ is typically annealed over time, starting with a high value and gradually decreasing it as the agent learns more about the environment. The approach used in this work is linear decay which aims to linearly decay ϵ from an initial value ϵ_{start} to a minimum value ϵ_{end} over a specified number of steps N. Eq. (9) shows the exploration rate for each time step t.

$$\epsilon_t = \max\left(\epsilon_{\text{end}} \ , \epsilon_{\text{start}} \ \frac{\epsilon_{\text{start}} \ - \epsilon_{\text{end}}}{N}.t\right) \tag{9}$$

where ϵ_t is the exploration rate at step t. ϵ_{start} is the initial exploration rate (typically 1.0).
ϵ_{end} is the minimum exploration rate (e.g., 0.1). N is the total number of steps over which ϵ is decayed. t is the current step.

iii) **Training FPS (Frames Per Second):** This measures the number of frames (or timesteps) processed per second during training. It indicates the efficiency of the training process. Higher FPS values mean the agent is able to learn faster, which is especially important for environments with a large number of frames**.** If F is the total number of frames processed and T is the total training time in seconds, the Training FPS is calculated as seen in Eq. (10) below:

$$FPS = \frac{F}{T} \tag{10}$$

where F is the total number of frames, and T is the total training time in seconds.

iv) **Training Learning Rate:** This is the step size used during the optimization process to update the weights of the neural network. It affects the speed and stability of learning. A learning rate that is too high can cause the training process to become unstable, while a learning rate that is too low can make the training process very slow**.**

v) **Training Loss:** This is the value of the loss function, which measures the difference between the predicted output and the actual target. It indicates how well the neural network is learning. Lower training loss generally means that the model is making more accurate predictions. In the context of reinforcement learning, this often involves minimising the difference between predicted Q-values and target Q-values or minimising the policy gradient loss.

The framework comprises a set of linked modules that collaborate to simulate RF spectrum situations. The RFRL Gym modules adhere to the Open AI Gym API, ensuring compatibility with third-party gym-compatible RL software. They include specified functions and methods that facilitate communication between a Reinforcement Learning agent and the environment. Consequently, the RFRL Gym utilises the "RL Cycle" training protocol to its advantage. This protocol involves the exchange of information between the Gym and RL agent in a cyclical manner. The exchange occurs across a specific number of training episodes, with each episode consisting of several time steps. The cycle begins at the start of each time step as the RL agent does an action within the custom-designed RF environment. This action might involve either occupying a channel or refraining from transmitting in the subsequent time step. At this juncture, Non-RL Entities will likewise choose an action. The acts of the Non-RL Entities are used to assess and award a reward to the action taken by the RL Agent. The RL Agent is provided with access to both this incentive and the newly expanded observation space for training purposes. This technique is repeated for each individual timestep.

The environment comprises 10 wireless channels and learning episodes that each consist of 100 time steps. The training process consists of 100 episodes and is run over 1000 epochs. Other training parameters are captured in Table II. A Q- learning agent is utilised, which balances exploration and exploitation using an epsilon-greedy strategy.

Fig. 3 shows a visualisation of this cognitive Radio network. Columns 0 to 9 represent the number of channels, and the rows refer to the number of timesteps during learning. When the agent makes an optimal channel choice, it shows green if there is no collision with other entities, and the red signal signifies collision with other entities in the environment.

- **Scenario: Single Entity DSA**

The environment in which the agent learns is defined by a set of parameters, known as a scenario. This particular scenario is designed to be straightforward, allowing for the validation of the RFRL Gym's performance and the accuracy of its results. It comprises two (2) constant entities which represent the primary users and twenty (20) fixed hopping entities which represents secondary users.

This simple scenario is designed to test the performance and accuracy of the RFRL Gym environment. The fixed hopping entity moves through all channels non-sequentially, while the two constant entities occupy channels 0 and 9. This setup demonstrates the multi-entity capability of the environment. As shown in Fig. 3, the Q-Learning agent can successfully navigate and avoid multiple entities, achieving an optimal policy. The RFRL Gym's gamified rendering functionality allows for easy visualization of the agent's optimal policy. The absence of collisions after convergence, as opposed to before (see Fig. 4), is clearly evident in Fig. 5. A key finding is that a Q-Learning agent in DSA mode can achieve an optimal policy when entities exhibit a stationary distribution, meaning the probability of channel selection remains constant.

i) Model Training and Evaluation

Two models are trained using the Deep Q-Networks (DQN) and Quantile Regression (QR-DQN) algorithms. The performance of both models are evaluated in subsequent sections. As expected, Q-Learning agents were able to converge to an optimal policy quickly within the 100 episodes. Figs. 4 and 6 show the agent's performance before training using the DQN and QR-DQN algorithms, respectively. Figs. 5 and 7 visualise the render results after learning. Both the DQN and QR-DQN agents avoid collision with other entities within the environment successfully.

ii) Training Results

Tables III and IV provides a summary of the performance of the DQN and QR-DQN algorithms respectively. The summary covers their performance after training, starting from 10,000 steps and increasing incrementally by 10,000 steps until reaching the final step of 100,000. The results are then evaluated in three phases: the initial phase (10,000 – 30,000 steps), the middle phase (40,000 – 70,000 steps) and the final phase (80,000 – 100,000 steps).

DQN - initial phase (10,000 - 30,000 steps)

The mean episode reward starts at 22.11 at 10,000 steps and increases steadily to 39.50 by 30,000 steps, exploration rate decreases from 0.8996 to 0.6988 (a high initial exploration rate is typical, allowing the agent to explore the environment and gather sufficient information) and train loss decreases from 0.0888 to 0.0429 indicating that the model is effectively minimizing the error between predicted and actual rewards. The learning rate remains constant at 1.00E-04 throughout the training process.

DQN - middle phase (40,000 - 70,000 steps)

The reward continues to increase, reaching 71.88 by 70,000 steps. The improvement in rewards suggests that the DQN model is consistently learning the optimal policy. The exploration rate continues to decrease, reaching 0.2972, pointing to the fact that the model is gradually shifting from exploration to exploitation, relying more on learned policies to select channels. The train loss fluctuates, with a slight increase at 50,000 steps (0.1277) and 60,000 steps (0.1867), followed by a rise at 70,000 steps (0.2181). The

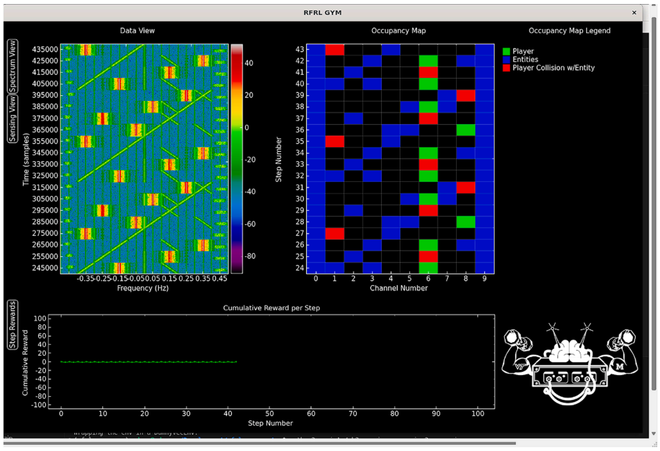


Fig. 3. Cognitive Radio Network Visualization. Illustrates the cognitive radio network with channels (columns 0–9) and timesteps (rows) during learning. Green indicates optimal channel selection with no collision, while red signifies collisions with other entities in the environment.

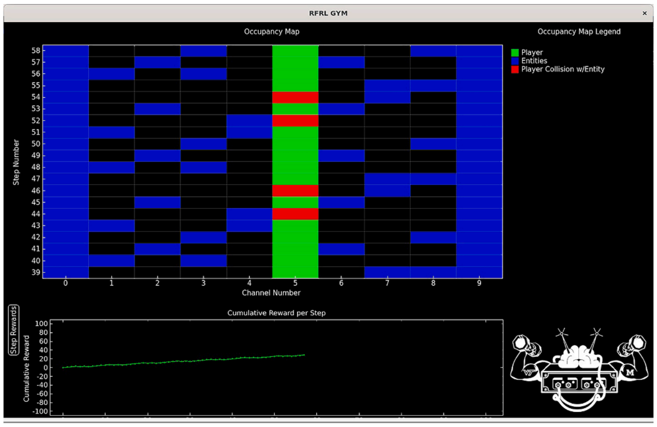


Fig. 4. Agent Performance Before Training Using the DQN Algorithm. Displays the agent's initial performance prior to training with the DQN algorithm, highlighting baseline behaviour in the cognitive radio network environment.

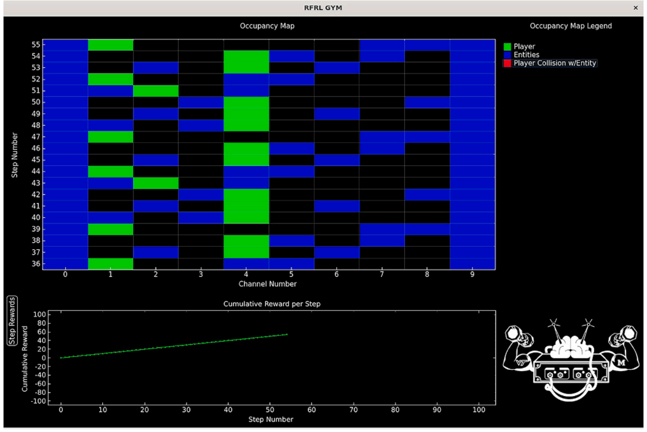


Fig. 5. Render Results After Learning. Visualizes the render results post-learning, showing how the DQN agent successfully avoid collisions with other entities in the environment.

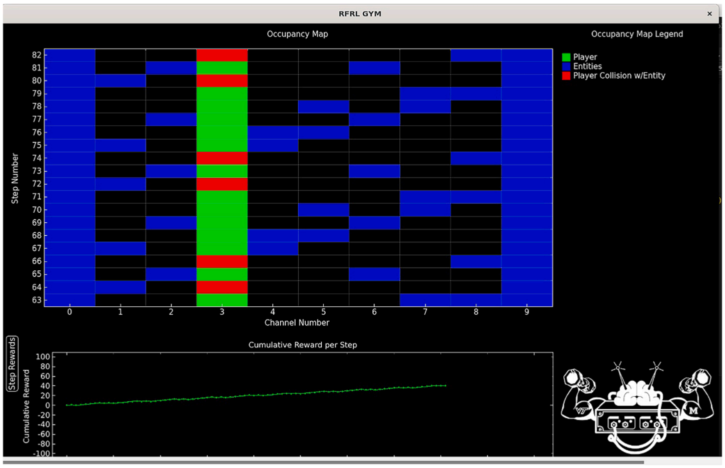


Fig. 6. Agent Performance Before Training Using the DQN Algorithm. Displays the agent's initial performance prior to training with the QR-DQN algorithm, highlighting baseline behaviour in the cognitive radio network environment.

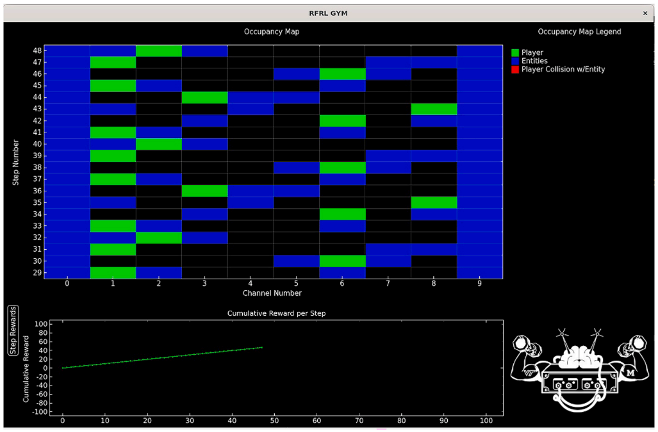


Fig. 7. Render Results After Learning. Visualizes the render results post-learning, showing how the QR-DQN agent successfully avoid collisions with other entities in the environment.

Table III
DQN performance.

Step	Mean Episode Reward	Explorati-on Rate	Time fps	Train Loss
10,000	22.11	0.8996	455	0.0888
20,000	29.80	0.7992	417	0.0725
30,000	39.50	0.6988	389	0.0429
40,000	47.07	0.5984	411	0.0388
50,000	55.14	0.4980	426	0.1277
60,000	62.84	0.3976	436	0.1867
70,000	71.88	0.2972	427	0.2181
80,000	79.52	0.1968	419	0.2648
90,000	88.60	0.0964	414	0.0145
100,000	96.34	0.0010	408	0.2843

Table IV
QR-DQN performance.

Step	Mean Episode Reward	Exploration Rate	Time fps	Train Loss
10,000	19.18	0.8996	308	3.8899
20,000	24.20	0.7992	299	7.2957
30,000	39.34	0.6988	302	6.6512
40,000	46.77	0.5984	300	5.7857
50,000	53.83	0.4980	307	5.1813
60,000	64.41	0.3976	314	4.8185
70,000	71.86	0.2972	299	2.9729
80,000	79.96	0.1968	295	3.5596
90,000	88.83	0.0964	295	4.1140
100,000	95.97	0.0010	293	3.4120

fluctuations in train loss suggest that the model might be encountering more complex decision scenarios, leading to temporary increases in loss.

DQN - final phase (80,000 - 100,000 steps)

The reward reaches its peak at 96.34 by 100,000 steps, indicating that the model has likely converged to a near-optimal policy. The exploration rate drops to near zero (0.0010) by 100,000 steps. At this stage, the model is primarily exploiting the learned policy, which suggests it has sufficient knowledge of the environment to make confident decisions. The train loss continues to fluctuate, with a notable low at 90,000 steps (0.0145) before rising to 0.2843 by 100,000 steps. The increase in loss towards the end indicates overfitting or difficulty in fine-tuning the final policy. However, the low loss at 90,000 steps also suggests the model had moments of very accurate learning.

QR-DQN - initial phase (10,000 - 30,000 steps)

The mean episode reward begins at 19.18 at 10,000 steps and increases to 39.34 by 30,000 steps, exploration rate starts at 0.8996 and decreases to 0.6988. The train loss is initially high at 3.8899 and increases significantly to 7.2957 by 20,000 steps before slightly decreasing to 6.6512 by 30,000 steps. This high train loss early on shows that the model is struggling with the learning process, possibly due to the complexity of the environment.

QR-DQN - middle phase (40,000 - 70,000 steps)

The reward continues to grow, reaching 71.86 by 70,000 steps, suggesting the model is successfully refining its policy and achieving higher rewards as it progresses. The exploration rate also continues to decrease, reaching 0.2972. The train loss decreases progressively, reaching 2.9729. The decreasing trend in train loss indicates that the model gradually improves its predictions, reducing the error over time.

QR-DQN - final phase (80,000 - 100,000 steps)

The reward peaks at 95.97 by 100,000 steps, indicating that the model has likely converged to a near-optimal policy. The pattern is similar to the DQN model, showing that the QR-DQN also effectively learns to optimize channel selection. Similarly, the exploration rate drops to near zero (0.0010). The train loss fluctuates slightly, with a low of 3.4120. While the fluctuations in train loss suggest that the model encounters varying levels of difficulty in learning, the overall trend indicates improvement, with lower losses compared to the early phase.

Interpretation and analysis

Mean episode reward

Fig. 8 captures the mean episode reward of the trained models. Both models achieve similar mean episode rewards, with DQN slightly outperforming QRDQN by a marginal difference, especially at the early stages of training. Both agents have similar results from 60,000 steps. This indicates that both algorithms are capable of learning and making decisions that result in high rewards in the environment.

Exploration rate

As seen in Fig. 9, both models start with a high exploration rate of 0.90 and gradually decrease it over time as they learn. A decreasing exploration rate implies that the agents are transitioning from exploration (trying out different actions) to exploitation (leveraging learned knowledge to maximize rewards).

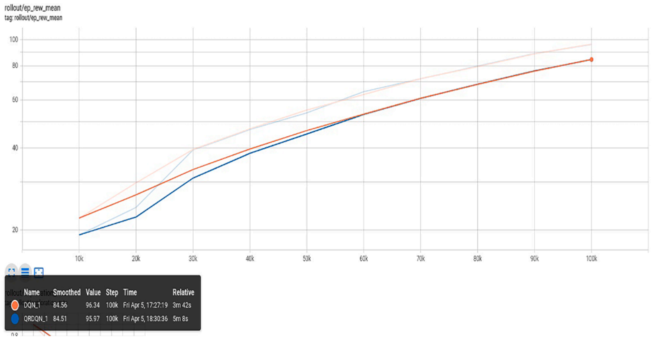
Training FPS (frames per second)

Fig. 10 shows that DQN consistently achieves a higher training FPS (around 400) compared to QRDQN (around 300). Higher FPS indicates faster training, suggesting that the DQN algorithm converges faster or require fewer training iterations to achieve similar performance compared to QRDQN.

Training learning rate

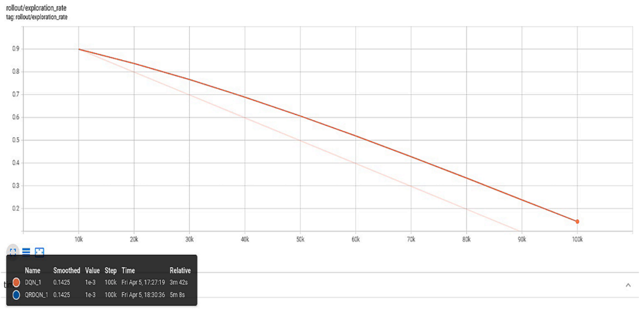
Both models utilize a small learning rate throughout training (1.00E-04 for DQN and 5.00E-05 for QRDQN). This ensures that the models update their parameters gradually, preventing drastic changes that may disrupt learning stability.

A learning rate of 0.0001 is relatively small, which helps ensure that the model learns at a stable pace. Since DQN relies on Q-value updates through backpropagation in a deep neural network, a high learning rate can cause the model to overshoot optimal values, resulting in instability or divergence in learning. A smaller learning rate like 0.0001, as used for this training, provides smoother updates to the Q-values, leading to more stable training. QR-DQN uses quantile regression to approximate the distribution of Q-values, and this requires more precise adjustments to the quantile estimates compared to standard Q-value estimates in DQN. Using a smaller



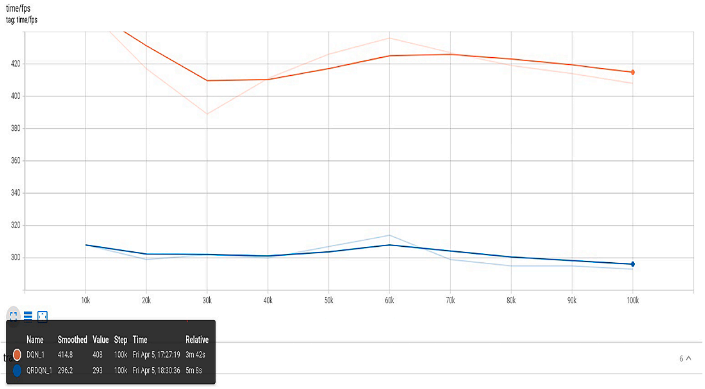
	Name	Smoothed	Value	Step	Time	Relative
●	DQN_1	84.56	96.34	100k	Fri Apr 5, 17:27:19	3m 42s
●	QRDQN_1	84.51	95.97	100k	Fri Apr 5, 18:30:36	5m 8s

Fig. 8. Mean Episode Reward of Trained Models Shows the mean episode reward for the trained models, with DQN slightly outperforming QRDQN.



	Name	Smoothed	Value	Step	Time	Relative
●	DQN_1	0.1425	1e-3	100k	Fri Apr 5, 17:27:19	3m 42s
●	QRDQN_1	0.1425	1e-3	100k	Fri Apr 5, 18:30:36	5m 8s

Fig. 9. Exploration rate. Illustrates the exploration rate progression for both models during training.



	Name	Smoothed	Value	Step	Time	Relative
●	DQN_1	414.8	408	100k	Fri Apr 5, 17:27:19	3m 42s
●	QRDQN_1	296.2	293	100k	Fri Apr 5, 18:30:36	5m 8s

Fig. 10. Training Frames per second. Shows FPS differences between both models.

learning rate like 0.00005 helps prevent large fluctuations in quantile estimates, ensuring the model learns the distribution more accurately (Fig. 11).

Training loss

DQN exhibits a lower training loss overall, fluctuating between 0.01 and 0.28. QRDQN shows a higher training loss, fluctuating between 2.97 and 7.30. Lower training loss typically indicates that the model is converging well and effectively minimizing the discrepancy between predicted and actual values (Fig. 12).

Performance benchmarking

To quantitatively assess the performance of the DQN and QR-DQN models in terms of throughput, latency, and interference avoidance in an environment with 10 channels, 2 primary users, and 20 secondary users, the provided metrics and general knowledge of QR-DQN performance in cognitive radio networks are employed.

i) Throughput

Throughput is the rate at which successful data transmissions occur over the network, typically measured in bits per second (bps) or packets per second (pps). Since both models achieve similar mean rewards, we can expect them to exhibit comparable throughput. Each reward point corresponds to a successful packet transmission [28]. Both models at 0.9634 and 0.9597 packets per time-step perform better than the GFMA approach in [29].

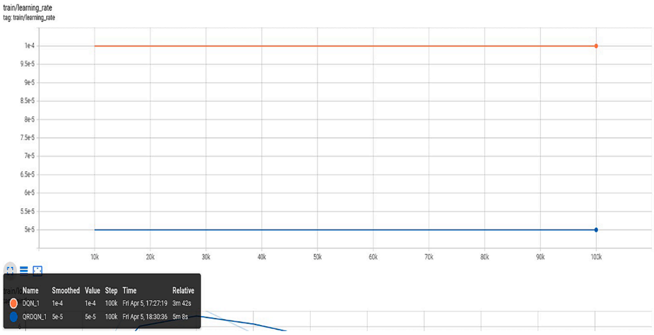
Mean Episode Reward: DQN - 96.34 QRDQN - 95.97

Time-Steps per Episode: 100

Throughput can be calculated as:

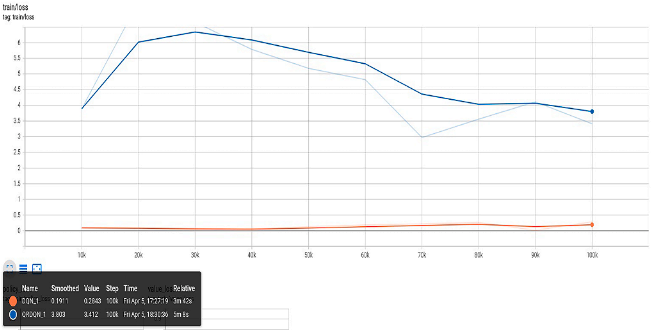
$$DQN = \frac{96.34\ \text{packets}}{100\ \text{timesteps}} = 0.9634\ \text{packetsper}\ \ \text{time} - \text{step} \tag{11}$$

$$QRDQN = \frac{95.97\ \text{packets}}{100\ \text{timesteps}} = 0.9597\ \text{packetspertime}\ - \text{step} \tag{12}$$



	Name	Smoothed	Value	Step	Time	Relative
●	DQN_1	1e-4	1e-4	100k	Fri Apr 5, 17:27:19	3m 42s
●	QRDQN_1	5e-5	5e-5	100k	Fri Apr 5, 18:30:36	5m 8s

Fig. 11. Training Learning Rate. Shows the learning rates used by both models (0.0001 for DQN) and (0.00005 for QRDQN).



	Name	Smoothed	Value	Step	Time	Relative
●	DQN_1	0.1911	0.2843	100k	Fri Apr 5, 17:27:19	3m 42s
○	QRDQN_1	3.803	3.412	100k	Fri Apr 5, 18:30:36	5m 8s

Fig. 12. Training Loss. Shows training loss values as models are trained.

ii) Latency

Latency is the time delay experienced in the system, often measured as the time taken for a packet to travel from the source to the destination [30]. Each time-step corresponds to one frame, and one frame represents a fixed transmission period.

Exploration Rate: Decreasing from 0.90 to 0.001

Training FPS(frames per second): DQN – 400 QRDQN - 300

Average latency per packet can be approximated by the duration of each frame.

$$DQN = \frac{1\text{ second}}{400\text{ frames}} = 2.5\text{ millisecondsperpacket} \tag{13}$$

$$QRDQN = \frac{1\text{ second}}{300\text{ frames}} = 3.33\text{ millisecondsperpacket} \tag{14}$$

iii) Interference Avoidance

Interference avoidance refers to the system's ability for optimal channel selection, minimize collisions and interference from other users, leading to more successful transmissions [31]. The environment has high interference potential due to the presence of 2 primary users and 20 secondary users. Interference avoidance can be quantified by the successful transmission rate and the reduction in collision rate: The DQN model successfully avoids interference 96.34 % of the time compared to 95.97 % of QRDQN, this implies a high level of interference avoidance efficiency.

Interference avoidance rate:

$$DQN = \frac{96.34\text{ successfultransmissions}}{100\text{ totaltransmissions}} * 100 = 96.34\% \tag{15}$$

$$\text{QRDQN} \ = \frac{95.97\text{ successfultransmissions}}{100\text{ totaltransmissions}} * 100 = 95.97\% \tag{16}$$

These results demonstrate the effectiveness of these deep reinforcement learning algorithms in managing spectrum access and avoiding interference with licensed users. This ensures reliable communication and optimal use of available spectrum resources, which is crucial for maintaining quality of service in dynamic and crowded spectral environments. The low average latency values of 1 millisecond for DQN and 3.33 milliseconds for QR-DQN per packet indicate that these models can operate efficiently in real-time scenarios, supporting applications that require prompt and reliable data transmission. This is particularly important for time-sensitive applications such as emergency communications, live broadcasting, and critical infrastructure monitoring.

Both DQN and QRDQN models demonstrate promising performance based on the evaluation metrics. The similar mean episode rewards suggest that both models are effective in learning optimal strategies for channel optimization in a cognitive radio network environment. However, DQN shows slightly better training efficiency with higher FPS and lower training loss.

Despite QR-DQN's theoretical advantages, such as capturing reward distributions and better handling uncertainty, DQN often achieves superior performance due to its simplicity, robustness, and stability. DQN's focus on optimizing expected rewards makes it well-suited for tasks like channel optimization in cognitive radio networks, where mean rewards are sufficient for decision-making. In contrast, QR-DQN's additional computational complexity, sensitivity to hyperparameters, and potential instability during training can hinder its performance. Moreover, QR-DQN's higher training loss in some phases may indicate difficulties in fitting reward distributions or overfitting to noise.

In a real environment, the choice between DQN and QRDQN would depend on factors such as computational resources, training time, and specific characteristics of the problem domain. If faster training and lower computational resources are priorities, DQN is preferred. If the problem requires more sophisticated action-value estimation and the computational cost is not a concern, QRDQN is a better choice.

Conclusion

This research work was carried out with the aim of contributing to the ongoing efforts to optimize spectrum utilization in the face of increasing wireless communication demands. The integration of Deep Reinforcement Learning into cognitive radio networks for DSA promises to enhance the efficiency, adaptability, and intelligence of spectrum sharing, enabling innovation in wireless communication technologies. We have compared the performance of two different agents using DQN and QR-DQN algorithms. Simulations have shown that the DQN can achieve a near-optimal decision accuracy in this system scenario even without the prior knowledge of the system dynamics Additionally, DQN has lower computational complexity, however, if the problem requires more sophisticated action-value estimation and the computational cost is not a concern, QRDQN is a better choice. For a more realistic simulation, the hardware functionality can be developed such that signals experience realistic channel conditions in which fading and multi-path propagation exist. Hybrid models that combine the strengths of DQN, QR-DQN and/or other RL algorithms can be explored. For example DQN can be used for initial fast training and QR-DQN can be used for fine-tuning and stability.

CRediT authorship contribution statement

Udeme C. Ukpong: Software, Formal analysis, Visualization, Writing – original draft. **Olabode Idowu-Bismark:** Supervision, Writing – review & editing. **Emmanuel Adetiba:** Conceptualization, Resources, Funding acquisition, Validation. **Jules R. Kala:** Methodology, Formal analysis. **Emmanuel Owolabi:** Project administration. **Oluwadamilola Oshin:** Writing – review & editing. **Abdultaofeek Abayomi:** Writing – review & editing. **Oluwatobi E. Dare:** Writing – original draft.

Declaration of competing interest

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

Acknowledgments

The authors acknowledge the Covenant Applied Informatics and Communication Africa Centre of Excellence (CApICACE) for sponsoring this research project. Additionally, the authors also acknowledge Google for providing part funding to the first author through the Google Award for TensorFlow Outreaches in Colleges, which was granted to the third author (EA).

References

[1] H. Mazar, Radio spectrum management: policies, regulations and techniques, Radio Spectr. Manag. Polici. Regul. Tech. (2016) 1–420, https://doi.org/10.1002/9781118759639.

[2] J. Notcker, E. Adetiba, K.K. Ronoh, A. Abayomi, O.A. Akinola, A spectrum sensing and allocation model for primary user detection and interference mitigation in television whitespaces, J. Comput. Sci. 20 (3) (2024) 333–343, https://doi.org/10.3844/jcssp.2024.333.343.

[3] A.H. Ifijeh, et al., Exploring television white space as an alternative for wireless broadband connectivity, in: 2023 2nd International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS), IEEE, 2023, pp. 1–7.

[4] I. Bala, K. Ahuja, K. Arora, D. Mandal, A comprehensive survey on heterogeneous cognitive radio networks, Compr. Guid. to Heterog. Networks (2023) 149–178.

[5] A. Marouthu, V. Srikanth, S. Sandeep, M.J. Babu, D. Haritha, Cognitive radio networks with reinforcement learning algorithms for spectrum allocation: a survey, Int. J. Adv. Trends Comput. Sci. Eng. 9 (5) (2020) 8371–8384, https://doi.org/10.30534/ijatcse/2020/211952020.

[6] M. Hu, The Art of Reinforcement Learning, 2023, https://doi.org/10.1007/978-1-4842-9606-6.

[7] F. Slimeni, Z. Chtourou, A. Ben Amor, Reinforcement learning based anti-jamming cognitive radio channel selection, Proc. Int. Conf. Adv. Syst. Emergent Technol. IC_ASET 2020 (2020) 431–435, https://doi.org/10.1109/IC_ASET49463.2020.9318287.

[8] H. Zhang, N. Yang, W. Huangfu, K. Long, V.C.M. Leung, Power control based on deep reinforcement learning for spectrum sharing, IEEE Trans. Wirel. Commun. 19 (6) (2020) 4209–4219.

[9] F. Tang, Y. Zhou, N. Kato, Deep reinforcement learning for dynamic uplink/downlink resource allocation in high mobility 5G HetNet, IEEE J. Sel. Area. Commun. 38 (12) (2020) 2773–2782.

[10] Y. Huang, C. Xu, C. Zhang, M. Hua, Z. Zhang, An overview of intelligent wireless communications using deep reinforcement learning, J. Commun. Inf. Networks 4 (2) (2019) 15–29, https://doi.org/10.23919/JCIN.2019.8917869.

[11] N.C. Luong, et al., Applications of deep reinforcement learning in communications and networking: a survey, IEEE Commun. Surv. Tutor. 21 (4) (2019) 3133–3174, https://doi.org/10.1109/COMST.2019.2916583.

[12] E. Adetiba, O.T. Ajayi, J.R. Kala, J.A. Badejo, S. Ajala, A. Abayomi, LeafsnapNet: an experimentally evolved deep learning model for recognition of plant species based on leafsnap image dataset, J. Comput. Sci. 17 (3) (2021) 349–363, https://doi.org/10.3844/JCSSP.2021.349.363.

[13] N.A. Khalek, D.H. Tashman, W. Hamouda, Advances in machine learning-driven cognitive radio for wireless networks: a survey, IEEE Commun. Surv. Tutorials (2023).

[14] M.C. Hlophe, A model-based deep learning approach to spectrum management in distributed cognitive radio networks, University of Pretoria (South Africa) (2020).

[15] O. Naparstek, K. Cohen, Deep multi-user reinforcement learning for distributed dynamic spectrum access, IEEE Trans. Wirel. Commun. 18 (1) (2019) 310–323, https://doi.org/10.1109/TWC.2018.2879433.

[16] A. Hazra, V.M.R. Tummala, N. Mazumdar, D.K. Sah, M. Adhikari, Deep reinforcement learning in edge networks: challenges and future directions, Phys. Commun. (2024) 102460.

[17] Y. Bai, H. Zhao, X. Zhang, Z. Chang, R. Jäntti, K. Yang, Towards autonomous multi-UAV wireless network: a survey of reinforcement learning-based approaches, IEEE Commun. Surv. Tutorial. (2023).

[18] U. Ukpong, O. Idowu-Bismark, E. Adetiba, O. Dare, E. Owolabi, R.J. Kala, Deep reinforcement learning applications for coexistence in television whitespace: a mini-review, in: 2024 International Conference on Science, Engineering and Business for Driving Sustainable Development Goals (SEB4SDG), 2024, pp. 1–9, https://doi.org/10.1109/SEB4SDG60871.2024.10629684.

[19] Y. Li, W. Zhang, C.X. Wang, J. Sun, Y. Liu, Deep reinforcement learning for dynamic spectrum sensing and aggregation in multi-channel wireless networks, IEEE Trans. Cogn. Commun. Netw. 6 (2) (2020) 464–475, https://doi.org/10.1109/TCCN.2020.2982895.

[20] F. Restuccia, T. Melodia, J. Ashdown, Spectrum challenges in the internet of things: state of the art and next steps, IoT Def. Natl. Secur. (2022) 353–375.

[21] G. Omondi, T.O. Olwal, Towards artificial intelligence-aided MIMO detection for 6G communication systems: a review of current trends, challenges and future directions, *e-Prime - Adv. Electr. Eng. Electron. Energy* 6 (November) (2023), https://doi.org/10.1016/j.prime.2023.100376.

[22] S. Wang, H. Liu, P.H. Gomes, B. Krishnamachari, Deep reinforcement learning for dynamic multichannel access in wireless networks, IEEE Trans. Cogn. Commun. Netw. 4 (2) (2018) 257–265, https://doi.org/10.1109/tccn.2018.2809722.

[23] S. Liu, X. Hu, W. Wang, Deep reinforcement learning based dynamic channel allocation algorithm in multibeam satellite systems, IEEE Access 6 (2018) 15733–15742, https://doi.org/10.1109/ACCESS.2018.2809581.

[24] Y. Yu, T. Wang, S.C. Liew, Deep-reinforcement learning multiple access for heterogeneous wireless networks, IEEE J. Sel. Area. Commun. 37 (6) (2019) 1277–1290, https://doi.org/10.1109/JSAC.2019.2904329.

[25] D. Rosen, et al., RFRL gym: a reinforcement learning testbed for cognitive radio applications, in: *Proc. - 22nd IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2023*, 2023, pp. 279–286, https://doi.org/10.1109/ICMLA58977.2023.00046.

[26] Independent Communications Authority of South Africa, "Reference Geolocation Spectrum Database." Accessed: Apr. 04, 2024. [Online]. Available: https://tvwhitespaces.icasa.org.za/public/spectrum/list.

[27] M. Sewak, Deep Reinforcement Learning Frontier of AI (2019).

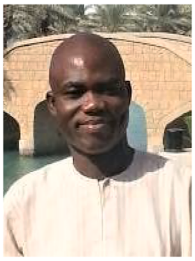
[28] Y. Wang, X. Li, P. Wan, R. Shao, Intelligent dynamic spectrum access using deep reinforcement learning for VANETs, IEEE Sens. J. 21 (14) (2021) 15554–15563, https://doi.org/10.1109/JSEN.2021.3056463.

[29] R. Huang, V.W.S. Wong, R. Schober, Throughput optimization for grant-free multiple access with multiagent deep reinforcement learning, IEEE Trans. Wirel. Commun. 20 (1) (2021) 228–242, https://doi.org/10.1109/TWC.2020.3024166.

[30] D. Chefrour, One-way delay measurement from traditional networks to sdn: a survey, ACM Comput. Surv. 54 (7) (2021) 1–35.

[31] S.A. Gbadamosi, G.P. Hancke, A.M. Abu-Mahfouz, Adaptive interference avoidance and mode selection scheme for D2D-enabled small cells in 5G-IIoT networks, IEEE Trans. Ind. Informat. 20 (2) (2023) 2408–2419.

Udeme Christopher Ukpong is a researcher at the Covenant Applied Informatics and Communication Africa Centre of Excellence (CApIC-ACE), a World Bank ACE-IMPACT Centre, and is currently pursuing a Ph.D degree in Information and Communications Engineering at Covenant University. He obtained his Master of Engineering (M.Eng) in Information and Communication Engineering at Covenant University and received a Bachelor's degree in Computer Engineering from the Kwame Nkrumah University of Science and Technology, Ghana in 2015. His research interests include machine intelligence, wireless communication, cognitive radio, cloud computing and High-performance computing. He can be contacted via email: udeme.ukpongpgs@stu.cu.edu.ng and udemechris1@gmail.com. https://orcid.org/0009-0005-4901-8601



Olabode Idowu-Bismark is a Senior Lecturer at Covenant University, Ota, Nigeria. He holds a B.Eng. degree in Electrical and Electronics Engineering from the University of Benin, Nigeria, and a M.*Sc.* in Telecommunications Engineering from Birmingham University UK. He obtained his Ph.D. in Information and Communication Engineering from Covenant University. Olabode has worked in various companies as an engineer, senior engineer, and technical manager. He is a member of the Nigerian Society of Engineers, a member, of MIEEE, and a COREN Registered Engineer. His research interest is in the area of mobile communication, mmWave, and MIMO communication. He has published many scientific papers in international peer-reviewed journals and conferences. He can be contacted at email: olabode.idowu-bismark@covenantuniversity.edu.ng. and idowubismarkolabode@gmail.com.

Emmanuel Adetiba (Member, IEEE) received the Ph.D. degree in information and communication engineering from Covenant Uni- versity, Ota, Nigeria. He was the Director of the Center for Systems and Information Ser- vices (aka ICT Center), Covenant University, from 2017 to 2019. He is the incumbent Deputy Director of the Covenant Applied Informatics and Communication Africa Centre of Excellence (CapIC-ACE) and a Co-PI of the FEDGEN Cloud Computing Research Project at the center (World Bank and AFD funded). He is the Founder and a Principal Investigator of the Advanced Signal Processing and Machine Intelligence Research (ASPMIR) Group. He is a Full Professor and the former Head of the Department of Electrical and Infor- mation Engineering, Covenant University, from 2021 to 2023. He is also an Honorary Research Associate (HRA) with the Institute of System Sciences and a Visiting Research Associate with the KZN e-Skills Co-Laboratory, Durban University of Technology, Durban, South Africa. He was a recipient of several past and ongoing scholarly grants from reputable bodies, such as World Bank; France Development Agency (AFD); Google; U.S. National Science Foundation; Durban University of Technology, South Africa; Nige- ria Communications Commission; Rockefeller Foundation; International Medical Informatics Association (IMIA); and hosts of others. He has authored/o-authored >100 scholarly publications in journals and conference proceedings, some of which are indexed in Scopus/ISI/CPCI. His research interests and experiences include machine intelligence, software- defined radio, cognitive radio, biomedical signal processing, and cloud and high performance computing (C&HPC). He is a registered engineer (R.Engr.) with the Council for the Regulation of Engineering in Nigeria (COREN) and a member of the Institute of Information Technology Pro- fessional (IITP), South Africa.

Jules Raymond Kala received a Ph.D. degree in Computer Science (focusing on image processing and pattern recognition) from the University of Kwazulu-Natal, South Africa, in 2017. He is an Assistant Professor of Data Science with the STEM faculty, at the International University of Grand Bassam. He is also a CIE consultant who designs AI solutions for electrical utilities. His broad research experiences and interests include Application development, data analytics, Artificial intelligence, Image processing, Mathematics, and Statistics. He enjoys teaching, research, and solving corporate-related issues using Artificial intelligence. He can be contacted at email: kala.j@iugb.edu.ci.

Emmanuel Owolabi holds a Bachelor's degree in Electrical Engineering from the University of Ilorin, Nigeria, and a Master's degree in Radio Communication and Signal Processing from Blekinge Tekniska Hogskolan (BTH) in Karlskrona, Sweden. He is a seasoned specialist in radio and digital communications with extensive experience in RAN and OSS implementations across various networks and OEMs in China, Sweden, and France. Amongst many firsts, he implemented the first 3 G commercial network project in Nigeria.He is an ISO-certified auditor through the International Register of Certificated Auditors (IRCA). He is an active member of several professional organizations, including IEEE, the Nigerian Society of Engineers (NSE), the Nigerian Institute of Electrical and Electronic Engineers (NIEEE), a Senior Member of the South African Institute of Electrical Engineers (SAIEE) and a member of the Advanced Signal Processing and Machine Intelligence Research (ASPMIR) Group.

Oluwadamilola Oshin received the Ph.D. degree in information and communication engineering (with a focus on nano-electronic biosensing) from Covenant University, Nigeria, in 2020. She is a Senior Lecturer of information and communication engineering with the Department of Electrical and Information Engineering, Covenant University. She is also a Faculty Member of the Covenant Applied Informatics and Communication Africa Centre of Excellence (CapIC-ACE), a World Bank ACE-IMPACT Centre, Covenant University. She has broad research experiences and interests including mobile communications, data analytics, artificial intelligence, and MEMS-based biosensing. She is professionally registered with the Council for the Regulation of Engineering in Nigeria (COREN) and is also a member of the Institute of Electrical and Electronics Engineers (IEEE). She enjoys teaching, research and solving health-related issues using engineering and technology. She can be contacted at email: damilola.adu@covenantuniversity.edu.ng.

Abdultaofeek Abayomi received the B.*Sc*. (Hons) Computer Science from the University of Ilorin, Nigeria. He later obtained Master of Technology in Computer Science from the Federal University of Technology, Akure. Nigeria. He has more than a decade stint with a financial solution service provider and an additional three years with an IT firm in Nigeria. He has lectured Computer Science, Information Technology and Information Systems courses at the Federal University Oye, Nigeria; Durban University of Technology, South Africa; Mangosuthu University of Technology, South Africa, respectively. He obtained a Ph.D. degree in Information Technology at the Durban University of Technology, Durban, South Africa and had postdoctoral research fellowship position at Mangosuthu University of Technology, South Africa. His research interests are artificial intelligence, machine learning and intelligence, wearable sensor technology, big data analytics, computer vision and image processing, bioinformatics, affective computing, human computer interaction (HCI), data mining, among others. He is a research associate with Walter Sisulu University, South Africa and a member of the Institute of Information Technology Professionals, South Africa (IITPSA) and the South African Institute of Computer Scientists and Information Technologist (SAICSIT).

Dare Oluwatobi Emmanuel received the B.Eng. degree in Electrical and Electronics Engineering from the University of Ilorin, Ilorin, Nigeria in 2019. He holds a Master of Engineering (M.Eng) degree in information and communication engineering from Covenant University, Ota, Nigeria and he is presently pursuing a Ph.D degree in information and communication engineering at the same institution. . He works as a Research Assistant with the Covenant Applied Informatics and Communication Africa Centre of Excellence (CapIC-ACE), a World Bank ACE-IMPACT Centre, Covenant University. His research interests include wireless communication, cloud computing, IoT and machine learning. He can be contacted via email: oluwatobi.darepgs@stu.cu.edu.ng and oluwatobidare61@gmail.com. https://orcid.org/0009-0002-4122-947X