UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

TÍTULO: Dynamic electromagnetic spectrum access through an action-specific deep recurrent Q-network

Trabajo de integración curricular presentado como requisito para la obtención del título de Ingenierio en Tecnologías de la Información

Autor:

Mateo Sebastián Lomas Olale

Tutor:

Manuel Eugenio Morocho Cayamcela, PhD.

Urcuquí, junio de 2023

Autoría

Yo, **MATEO SEBASTIAN LOMAS OLALE**, con cédula de identidad 1004295729, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el/la autor/a del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, junio de 2023.

Mateo Sebastián Lomas Olale

CI: 1004295729

Autorización de publicación

Yo, **MATEO SEBASTIAN LOMAS OLALE**, con cédula de identidad 1004295729, cedo a la Universidad de Investigación de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación

Urcuquí, junio de 2023.

Mateo Sebastián Lomas Olale

CI: 1004295729

Dedication

To my dearest parents and siblings, you have been the unwavering pillars of support in my life, and I am immensely grateful for your unconditional love and guidance that have helped shape me into the person I am today. Your countless sacrifices and unwavering encouragement have been a constant source of strength and motivation. This dedication is a small gesture of my immense love and gratitude to you.

Mateo Sebastián Lomas Olale

Acknowledgment

I am immensely grateful to my professor Manuel Eugenio Morocho, as well as to all the other teachers of the school, for their unwavering support, encouragement, and guidance that they have provided me throughout my academic journey. The mentorship and tutelage I have received from you all have been instrumental in molding me into who I am today. Your dedication and passion for teaching have been a great source of inspiration for me. I thank you for your invaluable contribution to my growth and development. I want to thank my dear friends Karol, Bolo, Nath, Martina, Nao, Darwin, Luis, and those who are not in our group anymore, but at some point, they were the best of my career. Also to my few classmates, Jhonny, Andres, and Juan. Your friendship has been a source of joy and laughter, and I am thankful for the memories we have created together. I want to thank Bri for being a fantastic person who has never left my side and has always been a constant source of support and positivity. You have changed my perspective on life through all the places we have visited. I owe you all. I cannot forget to thank my pets, who have been my loyal companions throughout my five years of school. Your unconditional love and companionship have brought me comfort and joy during both good and bad times.

Once again, I express my deepest appreciation to all who have contributed to my growth and success.

Mateo Sebastián Lomas

Resumen

La nueva generación de redes inalámbricas debe adaptarse al crecimiento del tráfico de datos móviles y soportar una demanda cada vez mayor de datos en dispositivos inalámbricos. Esto crearía una saturación de ocupación de espectro en pocos años a medida que se implementen tecnologías emergentes como 5G/6G. En trabajos recientes, investigadores han propuesto un nuevo método innovador de compartición de espectro llamado acceso dinámico al espectro (DSA) para resolver el problema de compartición de espectro entre bandas. Los usuarios secundarios (SUs) deben ser capaces de acceder a los huecos de espectro subutilizados de las bandas de usuarios primarios (PUs) para intentar llenar todo el sistema. Este acceso dinámico logra aumentar la eficiencia espectral y minimizar la interferencia entre los usuarios del espectro. En esta tesis, proponemos un enfoque de aprendizaje profundo de refuerzo basado en acciones, que es una modificación del algoritmo aprendizaje por refuerzo profundo. Nuestra propuesta introduce una capa intermedia de memoria a corto y largo plazo para recordar pares de acción-observación con el fin de tratar con canales no-observables. Nuestra red es capaz de aprender cómo acceder a canales inalámbricos de manera coordinada. La red puede recordar estados pasados y mantenerlos en su memoria, los cuales sirven para un mejor entrenamiento de la red. Evaluamos su rendimiento mediante simulaciones en escenarios casi parecidos a escenarios del mundo real. Los resultados muestran que nuestra estrategia propuesta puede superar a los esquemas basados en aprendizaje por refuerzo como el Q-learning para entornos de acceso dinámico. Estas mejoras incluyen: accesibilidad al canal y optimización en términos de memoria computacional. Esto permite a los SUs acceder a sus canales respectivos con una relación señal-ruido-interferencia óptima.

Palabras Clave:

Aprendizaje profundo, aprendizaje por refuerzo, interferencia, optimización de espectro,

comunicaciones inalámbricas.

Abstract

The new generation of wireless networks shall adapt to growth in mobile data traffic and support the increasingly high demand for data in wireless devices. This increase would create a spectrum occupancy saturation in a few years due to the implementation of emerging technologies like 5G/6G. Recently, researchers proposed a new innovative spectrum-sharing method called dynamic spectrum access (DSA) to solve the spectrum-sharing problem between bands. Secondary users (SUs), should be capable of accessing the underutilized spectrum holes of the primary users (PUs) bands to fulfill the whole system, increase the spectral efficiency, and minimize the interference between spectrum users. In this work, we proposed an action-specific deep reinforcement learning approach that modifies the deep reinforcement learning algorithm. Our proposal introduces an intermediate long-short-term memory layer for remembering action-observation pairs to deal with non-observable channels. Our network can learn how to access wireless channels in a coordinated way. The network mentioned above can remember past states and keep channels in its memory which is useful for improving the network training. We evaluate the performance of the network by conducting real-world scenario simulations. The results show that our proposal can overcome Q-learning-based schemes for DSA in terms of computational memory and channel accessibility which leads SUs to access their respective channels with an optimal signal-to-interference-plus-noise ratio.

Keywords:

Deep reinforcement learning, dynamic spectrum access, interference, spectrum optimization, wireless communications.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Wireless communications are the fastest-growing sector of the communication industry because of the invention of smartphones and the successful deployment of Wi-Fi and cellular networks. In the internet of things (IoT) era and with the increasing number of subscriptions, wireless traffic is increasing exponentially. The spectrum scarcity problem created by the static radio frequency (RF) allocation policy will worsen when millions of devices are connected to the wireless spectrum. Thus, dynamic spectrum access (DSA) using cognitive radio networks could solve the artificial spectrum scarcity problem by allowing unlicensed users to access licensed bands opportunistically [1]. Policy and technical challenges exist to realize DSA in cognitive radio networks fully. It is very challenging to adaptively allocate resources for DSA in cognitive radio networks for unlicensed secondary users while protecting primary users from harmful interference. For DSA, secondary users must identify spectrum opportunities (by sensing channels or searching them in a sensing database) to use them dynamically, see Figure 1.1. Once spectrum opportunities are identified, secondary users switch to transmission mode to communicate with their receivers. Dynamic spectrum access can be defined [2] as a mechanism to adjust the spectrum resource usage in a near-real-time manner in response to the changing environment and objective (e.g., available channel and type of applications), changes of radio state (e.g., transmission mode, battery status, and location) and changes in environment and external constraints (e.g., radio propagation, operational policy).

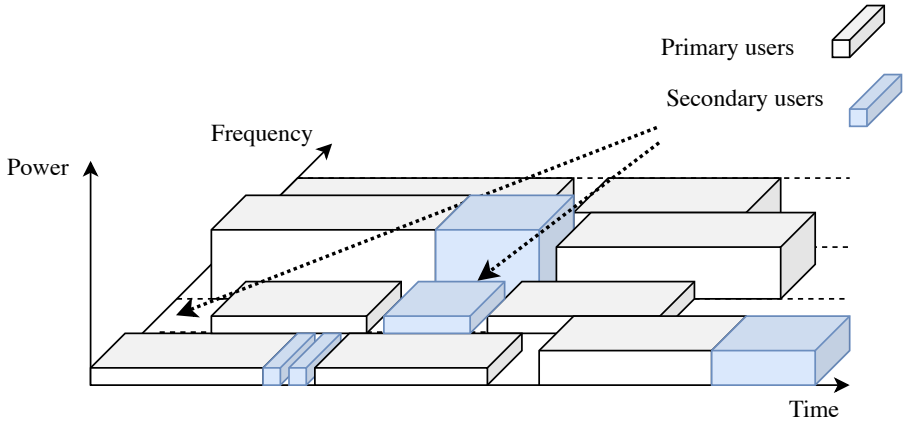

Figure 1.1: Spectrum Opportunity, a secondary user identifies a spectrum hole and tries to access it dynamically.

Many DSA schemes have been studied. One of them is opportunistic spectrum access which the SU (secondary users) sense or predict the spectrum holes of primary users (PUs) and accesses them dynamically. Ideally, the PUs are not aware of the presence of SUs, only in the case that SUs do not cause any interference; otherwise, it would detect a collision [3]. This scheme is efficient, but has some disadvantages, as it may suffer from interference, since it dynamically accesses and exchanges information with other channels. The objective of this is to predict the PUs' channel usage status (it can be idle or busy) and thus let SU access a channel, therefore, minimizing the interference ratio and maximizing the spectral utilization ratio. Another approach is the database-driven spectrum, where there is a database where we store spectrum availability built by propagation models or spectrum sensing measurements. This is effective but not highly recommended because of maintenance costs. Much artificial intelligence (AI) based schemes have been proposed to solve DSA drawbacks. One problem that DSA networks have is that SU does not know the state of the channel at a time slot. We can call it no observability because of sudden transmission changes in the environment. So, in this paper, we introduce action-specific deep recurrent Q-Network (ADRQN) architecture that outperforms previous methods, and we incorporate reinforcement learning.

1.2 Problem statement

As the scarcity of frequency spectrum is a major problem in the telecommunication and wireless fields, DSA has been identified as a promising solution to this problem. DSA allows the opportunistic access of licensed bands by unlicensed users without causing harmful interference to the licensed user. Actual artificial intelligence methods have been proposed, but they have many drawbacks. They do not take into account the large space in channels that a system can have, therefore leading to wrong results in simulations. A large space and real-world scenarios are necessary to reflect if the proposed solutions could work in these circumstances.

1.3 Objectives

1.3.1 General Objective

Implement a deep reinforcement algorithm that incorporates an action-specific network for dynamic spectrum access.

1.3.2 Specific Objectives

- Develop a decentralized dynamic spectrum access system based on deep reinforcement learning to find optimal channels for secondary users.
- Simulate clusters of users in a delimited geometry to reduce the computational load caused by Q-learning in large spaces.
- Maximize the channel allocation efficiency while ensuring reliability and maximizing the SINR and channel capacity.

Chapter 2

Theoretical Framework

2.1 Wireless Communications

The growth of wireless communication technology has been significant, positioning it as one of the fastest-growing segments within the communications industry. This trend has gained considerable attention from the media and the public, underlining its impact on the industry and society as a whole.

The proliferation of cellular systems, which have seen exponential growth over the last decade, is a testament to the increasing popularity of wireless communication. Currently, there are around two billion cellular phone users worldwide, making it a critical business tool and an integral part of everyday life in most developed countries. In addition, many developing countries are replacing outdated wireline systems with cellular networks, further driving the growth of wireless communication technology [2].

Wireless local area networks (WLANs) have also emerged as a popular alternative to wired networks, with many homes, businesses, and campuses currently supplementing or replacing their wired networks with WLANs. Moreover, advancements in wireless technology have given rise to new applications, such as wireless sensor networks, smart homes and appliances, automated highways and factories, and remote telemedicine, which were once mere research ideas but are now becoming concrete systems. The growth of wireless systems has been remarkable, with the potential for further expansion. However, certain technical challenges still persist that need to be tackled to ensure the robustness of wireless networks and the delivery of required performance to support emerging applications.

Hence, continuous research and development in wireless communication technology are imperative to overcome these challenges and ensure a promising future for wireless networks as standalone systems and as part of the larger networking infrastructure [1].

2.1.1 Key concepts

SINR

The signal-to-interference and noise ratio (SINR) is a metric used to quantify the strength of the desired communication signal relative to the power of both interference and noise. Here, SINR=S-I-N, where S represents the power of the signal, I represents the interference and N represents the noise. The measurement unit for SINR can be any relative dB unit, such as dBm or dBW, depending on the context of the application.

The main aim of a network operator is to optimize or maximize the SINR, as it allows for the highest modulation rates, throughput, and subsequently, the fastest and most reliable service to the end-user. A benchmark value can be established, where SINR $>$ 20dB is considered an excellent link, while SINR $<$ 0dB is considered an unusable link. The maximum possible bit rate and robustness of data transmission can be achieved by utilizing the highest Modulation and Coding Scheme (MCS), which is feasible only when the SINR value is greater than 20dB, according to [4].

WINNER II Model

The WINNER II channel model is a channel model for wireless communications based on extensive measurements and simulations of radio propagation in realistic urban and suburban environments. The model is intended to provide a realistic representation of the radio channel in these environments and is used to evaluate the performance of wireless communication systems in these scenarios [5]. The WINNER II channel model is developed by the Wireless World Research Forum (WWRF), and it covers different scenarios such as urban, suburban, and rural environments. Also, it covers different types of services such as cellular, wireless local area network (WLAN), and personal area network (PAN). The model includes parameters such as path loss, small-scale fading, shadowing, and multipath propagation

2.1.2 Dynamic Spectrum Access

The DSA technology enables radios to share frequency bands without disrupting other wireless systems in use. This strategy is achieved through a combination of RF, signal processing, networking, detection technologies, and DSA software algorithms. This enables significantly more capacity for communication than traditional static spectrum access methods. DSA enhances spectrum utilization in frequency, location, and time. It allows networks to utilize various frequencies at authorized and available times and places. If a non-cooperative user is detected on the same channel, DSA-enabled devices immediately switch to a vacant channel [6]. Multiple networks can share a spectrum band, as many frequencies are only used occasionally and in specific locations. Wireless service providers or spectrum users can deploy multiple applications or services within a single band with the help of DSA.

Cognitive Radio Network (CRN)

Cognitive Radio (CR) technology facilitates hierarchical coexistence in licensed spectrum bands based on interference avoidance or interference control paradigms [7]. This technology is cast as a solution to the spectrum under-utilization phenomenon resulting from the legacy spectrum licensing system based on command and control. License-exempt users, e.g., first responders, can intelligently access underutilized licensed spectrum bands using this technology. This underutilization occurs in the spatial dimension, i.e., location and the temporal dimension. CR technology can offer much more than just additional capacity. It can offer network resilience, flexible network topology, and security

Spectrum Mobility Management

The ultimate goal of spectrum mobility management is to perform successful and fast spectrum access while minimizing the interference with PUs [8]. Spectrum mobility management has four main functions:

- **Spectrum sensing**: SUs must monitor the available spectrum bands and then detect WSs. Spectrum sensing is an essential functionality in CRNs and is closely related to other spectrum management functions.

- **Spectrum decision**: Based on the spectrum sensing outcome, the spectrum decision procedure assigns the available channels to the SUs.
- **Spectrum sharing**: The role of spectrum sharing is to ensure fairness among SUs, especially when multiple SUs request access to the spectrum simultaneously.
- **Spectrum mobility**: The spectrum mobility procedure collaborates with the other three functionalities to detect the events that must initiate the spectrum evacuation process.

Spectrum analysis

In DSA environments, the widespread holes in the system have multiple characteristics that vary over time. Doing spectrum analysis enables the characterization of these given bands, which can be exploited to allocate an adequate spectrum band. Wireless environments in DSA are characterized by time-varying radio environment, the primary user activity, and band information such as operational frequency and bandwidth. The quality of a particular spectrum band can be described with the following parameters [9].

- **Interference**: Different spectrum bands have varying levels of congestion. As a result, the choice of spectrum band being used affects how much interference will be experienced in the channel. By analyzing the amount of interference that the primary receiver is exposed to, it is possible to determine the maximum power that can be used by a DSA user. This information is then used to estimate the channel capacity.
- **Wireless link errors**: Depending on the modulation scheme and the spectrum band's interference level, the channel's error rate changes.
- **Path loss**: as the operating frequency increases, the path loss also increases. As a result, if a DSA user maintains the same transmission power, their transmission range will decrease at higher frequencies. Conversely, if the transmission power is raised to counteract the increased path loss, it will cause more interference for other users [10].

- **Link layer delay**: Various link layer protocols are necessary to deal with distinct path losses, wireless link errors, and interference at different spectrum bands. As a consequence, there are varying transmission delays for link layer packets at each band.

- **Holding time**: If the holding time is longer, the quality of the connection will be improved. Frequent spectrum handoffs, however, can decrease the holding time. Therefore, when designing DSA networks that have a significant expected holding time, it is important to consider the previous statistical patterns of handoff.

The capacity of a channel, which can be determined by the aforementioned parameters, is the primary factor for spectrum characterization. Typically, capacity estimation has been based on the SNR observed by the receiver. Nonetheless, relying solely on SNR, which only considers the local observations of DSA users, is insufficient for preventing interference from primary users. Thus, spectrum characterization is focused on capacity estimation based on the interference at the licensed receivers. In [11], authors proposed a method for estimating spectrum capacity that takes into account the bandwidth and the maximum allowable transmission power. The equation used to estimate the spectrum capacity, denoted as C is given by Equation 2.1:

$$C = B \log \left(1 + \frac{S}{N + I} \right), \tag{2.1}$$

Here, B represents the bandwidth, S indicating the received signal power from the DSA user, N representing DSA receiver noise power, and I denoting the interference power received at the DSA receiver due to the primary transmitter.

Opportunity Identification

A fragment of the spectrum can be considered an opportunity if it is not occupied by primary users currently. There are different constraints to be satisfied to identify any channel as an opportunity. Suppose there are two secondary users, A and B, situated geographically far apart [12]. The presence of different primary users surrounding a DSA user is a crucial factor. If the DSA user can communicate over a channel without causing any interference to nearby primary users, then this channel can be considered an opportunity. If User A

desires to transmit to User B, A must take precautions to avoid interfering with nearby primary receivers, while B must protect nearby primary transmitters. In order to prevent interference, it is crucial to maintain a transmission-free zone within the range of RTx from A and the reception-free zone within the range of RRx from B, so that no primary user transmits or receives within those zones. RTx is dependent on the transmission power of A and relies on the primary users' interference tolerance limit, while RRx is determined by the transmission power of primary users and the secondary users' interference tolerance limit. The interference tolerance limit is a component of the regulations governing the spectrum. There are several crucial factors to consider when assessing spectrum opportunities, such as the geographical location of secondary users and primary users.

Opportunity Sharing

For secondary users to share the spectrum, coordination or control by a centralized entity is necessary. Therefore, the architecture of spectrum sharing can be categorized as either centralized or distributed. In a centralized architecture, all aspects of spectrum allocation are carried out by a central entity, while spectrum sensing is distributed and conducted by associated secondary users, as noted in [13] and [14]. Performing spectrum sensing and allocation in a proposed architecture requires cognitive capabilities in the secondary users. However, in the hybrid architecture, these tasks are carried out by a spectrum controller (SC), of which there are several in the network. This type of architecture is better suited to work with legacy systems since it does not rely on cognitive abilities from secondary users, as opposed to other architectures [15]. In a distributed architecture [16], spectrum sensing and allocation are carried out by individual nodes in a distributed manner. However, this method results in a lot of message exchanges between nodes, which is an overhead. Opportunity sharing in spectrum allocation requires secondary users to coordinate with each other and with primary users to avoid interference and efficiently use the available spectrum. Various factors, such as path loss, wireless link errors, interference, and primary user activity, affect the quality of the available channels. To estimate the channel capacity, a method that considers the bandwidth and permissible transmission power, as well as the received signal power, noise power, and interference power, can be used. The interference tolerance limit is an important aspect of spectrum

regulatory policy [17, 15] Then, efficient opportunity sharing requires careful consideration of various technical and regulatory factors and the selection of an appropriate spectrum sharing architecture.

2.1.3 Power Allocation

Power consumption is one of the major concerns in wireless communications between a fixed base station (BS) and several mobile terminals or for say mobile stations (MSs) interact within a certain coverage area, defined by the maximum distance beyond which the quality of the radio link becomes unacceptably poor. For a given transmission technique, this area is determined by the transmitted power level, the propagation medium, and by the receiver implementation. Portable wireless terminals are usually powered by batteries, and, therefore, transmission power is a scarce resource. In the basic design of a wireless system, the required transmitter power for a given coverage depends on the minimum received SNR, which guarantees a certain quality of service measured, e.g., in terms of bit error rate (BER). Hence, the required transmitter power from the MS can be reduced by a BS receiver design that corresponds to improved error performance, and, therefore, to lower requirements in terms of minimum acceptable SNR. Given a certain SNR threshold, adaptive RF power control is then often employed at both MS and BS to minimize the transmit power and reduce interference to co-channel users, while maintaining the quality of the radio link [18].

Power allocation is an effective technique for prolonging the lifetime of network terminals. Generally, optimum power allocation improves the efficiency of wireless systems. When power allocation is properly done, source information can reach the destination efficiently [19]. In recent years, scientists have conducted research on power allocation schemes based on cooperative communication. Phyla *et al.* [20] have proposed a power allocation scheme to minimize the transmitted power and the minimum loss of channel capacity as the goal for cooperative communication systems. Regarding power allocation based on channel capacity, Wang *et al.* [21] propose to maximize the capacity as the optimization goal, and presents the power allocation scheme based on the water filling algorithm, but the solution is only for two base stations did not involve two or more base stations of the cooperative communication situation. Xiao Hailin [22] proposes to maximize the capacity

as the optimization goal to solve the power allocation problem of cooperative communication systems, and formulates the comparison of the power allocation scheme based on a genetic algorithm and particle swarm optimization, but focuses only on the selection of the algorithms rather than an in-depth study on the power allocation scheme. In the aspect of power allocation based on error probability, Wu *et al.* [23] presents a minimum error probability power allocation scheme for cooperative communication systems, the contribution focuses on the analysis of the influence of the number of relays on the performance of the system. Optimum power allocation plays a significant role in ensuring that source information reaches the destination with high efficiency. Researchers have proposed various optimization goals, such as minimizing the transmitted power and channel capacity loss, maximizing capacity, and minimizing error probability. While some proposed power allocation schemes have limitations, such as being only applicable to two base stations or focusing only on the selection of algorithms, they nevertheless contribute to the ongoing efforts to optimize power allocation in cooperative communication systems.

2.2 Artificial Intelligence

The artificial intelligence algorithms can be classified depending on the learning approach that implements to achieve a certain object. The main categories are supervised learning, unsupervised learning, and reinforcement learning.

2.2.1 Artificial Neural Networks

Artificial neural networks (ANNs) are systems that are inspired by the human brain. They were initially proposed in the 1940s and 1950s [24] and saw significant developments in the 1980s due to the introduction of the backpropagation (BP) learning procedure [25]. ANNs function as parallel distributed computing networks and are composed of simple computational devices that are highly interconnected, much like biological neural networks. The connections between neurons in ANNs determine their function, and the weights of these connections are adaptive coefficients that determine the intensity of input signals. ANNs have been extensively used in various research fields to interpret complex and nonlinear phenomena in machine intelligence. While ANNs are not as complex as the human brain,

they share key similarities with their biological counterparts in terms of their building blocks and inter connectivity [26]. The inputs could be discrete or continuous data values, and likewise, the outputs also could be discrete or continuous. The input and output could also be deterministic or stochastic or fuzzy [27]. The mathematical representation of a neuron's working is detailed in equation 2.2

$$O = f\left(\sum_{i=1}^{n} w_i \cdot x_i + b\right) \tag{2.2}$$

Artificial neural networks typically consist of layers of processing units. The processing units within a layer exhibit similarity, with shared activation dynamics and output functions. Connections can be made either from the units of one layer to the units of another layer (interlayer connections) or among the units within the layer (intralayer connections) or both interlayer and intralayer connections. Further, the connections across the layers and among the units within a layer can be organized either in feedback manner or a feedforward manner. Here, in the feedback network the processing unit may be visited more than once [27]. There are several types of ANNs developed, but they can be broadly categorized into two main categories based on their learning process as follows.

2.2.2 Supervised learning

This learning approach is made with an input dataset and its corresponding target outputs. The main objective is to minimize the error between the network output and the target output. This process is done in the network's training step which consists of iterative computing and adjusting the ANN weights. Once the ANN produces a satisfactory output for the input, the training ends, and the weights are fixed so the network can be put into operation in the test phase [28].

2.2.3 Unsupervised learning

In this learning paradigm, the neural network is not provided with a target output; just the input dataset is given. The networks try to discover patterns or trends in the input data without an external teacher signal. This learning is also called self-organized learning since they establish a task-independent measure to evaluate the quality of representation

that the network is required to learn. Some common applications of unsupervised learning include clustering, pattern configuration, and principal components analysis [28].

2.2.4 Long short-term memory

LSTM, which is an abbreviation for long short-term memory, is a type of recurrent neural network (RNN) architecture that was designed to more accurately model temporal sequences and their extended dependencies than conventional RNNs [29].

The recurrent hidden layer of the LSTM architecture includes memory blocks that have memory cells with self-connections, which store the temporal state of the network. These memory blocks also consist of special multiplicative units known as gates that regulate the flow of information. Initially, each memory block in the LSTM architecture had an input gate and an output gate. The input gate regulated the inflow of input activations into the memory cell, while the output gate controlled the outflow of cell activations into the network. Subsequently, the forget gate was introduced to the memory block [30]. This improved the LSTM's capability to handle uninterrupted input streams that are not divided into sub-sequences by introducing the forget gate. The forget gate adjusts the cell's internal state before feeding it back to the cell through its self-recurrent connection, enabling it to adaptively erase or reset its memory. Also, the current design of LSTM incorporates peephole connections from the internal cells to the gates within the same cell. This enables the network to learn the precise timing of the outputs [31].

In an LSTM network, an input sequence $x = (x_1, \ldots, x_T)$ is mapped to an output sequence $y = (y_1, \ldots, y_T)$ by computing the network unit activations iteratively from $t = 1$ to T using the following equations:

$$\begin{aligned} i_t &= \sigma\left(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i\right) \\ f_t &= \sigma\left(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f\right) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g\left(W_{cx}x_t + W_{cm}m_{t-1} + b_c\right) \\ o_t &= \sigma\left(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o\right) \\ m_t &= o_t \odot h\left(c_t\right) \\ y_t &= \phi\left(W_{ym}m_t + b_y\right) \end{aligned} \tag{2.3}$$

The weight matrices are represented by the W terms, with W_{ix} indicating the matrix

of weights from the input gate to the input. The diagonal weight matrices for the peephole connections are denoted as W_{ic}, W_{fc}, and W_{oc}. The b terms indicate bias vectors, with b_i referring to the input gate bias vector. The logistic sigmoid function is represented by σ. The input gate, forget gate, output gate, and cell activation vectors are respectively denoted by i, f, o and c, with all of them being the same size as the cell output activation vector m. The element-wise product of the vectors is represented by $\odot$. The cell input and cell output activation functions are represented by g and h, respectively, with $tanh$ being the function used. The network output activation function, softmax, is represented by ϕ.

2.2.5 Reinforcement learning

Q-learning [32] is a representative reinforcement learning algorithm that learns the optimal policy in an interactive environment by trial and error. By assuming discrete time, in time slot k, the agent observes the states s_k of the environment and takes an action a_k based on a policy π. Upon the action being taken, the state moves from s_k to s_{k+1}, and the agent obtains a reward/cost k that indicates the benefit/loss by taking a_k at s_k. The optimal action policy π^* is computed by maximizing/minimizing the expectation of the future cumulative discounted reward/cost [33]. In Q-learning, a Q-function is defined to represent the expected future cumulative discounted reward for action a_k under state s_k. The values of the Q-function, i.e., Q-value, are stored in a Q-table, whose size is the number of states times the number of actions. The Q-value in time slot k is updated by the Equation below.

$$Q^{\text{new}}(s_t, a_t) \leftarrow (1-\alpha)\cdot \underbrace{Q(s_t, a_t)}_{\text{current value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \Big(\underbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}_{\text{new value (temporal difference target)}} \Big) \tag{2.4}$$

where r_t is the reward received when moving from the state s_t to the state s_{t+1}, and α is the learning rate ($0 < \alpha \leq 1$). Note that $Q^{new}(s_t, a_t)$ is the sum of three factors:

- $(1-\alpha)Q(s_t, a_t)$: the current value (weighted by one minus the learning rate)
- αr_t : the reward $r_t = r(s_t, a_t)$ to obtain if action a_t is taken when in state s_t (weighted

by learning rate)

- $\alpha\gamma \max_a Q(s_{t+1}, a)$: the maximum reward that can be obtained from state s_{t+1} (weighted by learning rate and discount factor)

An episode of the algorithm ends when state s_{t+1} is a final or terminal state. However, Q-learning can also learn in non-episodic tasks (as a result of the property of convergent infinite series). If the discount factor is lower than 1 , the action values are finite even if the problem can contain infinite loops. For all final states $s_f, Q\left(s_f, a\right)$ is never updated, but is set to the reward value r observed for state s_f. In most cases, $Q\left(s_f, a\right)$ can be taken to equal zero.

2.2.6 Deep learning

Deep learning is a powerful tool that uses multiple layers to analyze data with varying degrees of complexity. Its applications are vast and include speech recognition, object detection, and even drug discovery and genomics [34, 35, 36, 37]. These networks have multiple layers that process and transform input data, producing final output. During training, the network's weights are adjusted to minimize the difference between output and the predicted. Deep learning discover complex patterns in large data sets. It uses the back-propagation algorithm to guide the machine in adjusting its internal parameters, which are used to calculate the representation in each layer based on the representation of the previous layer.

Deep LSTM

Deep LSTM RNNs are designed to learn at varying time scales over input data by employing stacked LSTM layers, making them an effective type of neural network. These networks are deep architectures, resembling a feed-forward neural network unrolled in time, with each layer sharing the same model parameters [38]. The inputs to the model pass through multiple nonlinear layers. In contrast, the output contribution of the features from a particular time instance is made by a single nonlinear layer after processing. At each time step, the input goes through multiple LSTM layers and propagates through time and LSTM layers [38]. Through the use of multiple layers, deep LSTM networks are able

to allocate parameters more effectively across the network. Rather than increasing the memory capacity of a standard model by a factor of 2, it is possible to have four layers with nearly the same number of parameters. As a result, the inputs are subjected to a higher number of nonlinear operations for each time step.

2.2.7 Deep reinforcement learning

Deep reinforcement learning (DRL) combines deep neural networks with reinforcement learning algorithms to enable machines to learn from their own experience in complex and dynamic environments. DRL has shown promising results in various domains such as robotics, games, and autonomous driving [39]. By leveraging deep learning techniques and reinforcement learning principles, deep RL algorithms are capable of effectively handling massive amounts of input data, making informed decisions, and taking actions that optimize a specific objective (e.g, achieving the highest possible score in a game).

Deep Q-learning

Deep Q learning combines Q learning with deep neural networks that are used to approximate the Q function instead of storing it in a table. This allows deep Q learning to handle high-dimensional state spaces common in real-world environments. A neural network takes as input the current state and outputs a Q value for each possible action. The action with the highest Q value is chosen and the neural network is trained using a variant of Q learning called "temporal difference learning" [40] to find the Q value to update based on the observed reward. For instance, DQN uses a neural network parameterized by θ to represent $Q(s, a; \theta)$. By incorporating non-linear activation functions in the hidden layers and increasing the number of nodes, neural networks can approximate any function with an arbitrarily small error. DQN is optimized by minimizing the following loss function [41]:

where $y_i^{\text{target}} = r + \gamma \max_{a'} Q\left(s', a' \mid \theta_i^-\right)$ represents the target value of action a_t under a given state s_t. Here θ_i^- is cloned from θ_i every fixed number of iterations. DQN Store previous samples $e_t = (s_t, a_t, r_t s_{t+1})$ into fixed-size memory D_t using experience replay. The Q-network is trained using mini-batches of past experiences sampled uniformly from the replay memory. An important factor in the efficiency of DQNs on AlphaGo and Atari games is the assumption of full observability, which allows the DQN to take only few

observations as input [42]. Therefore, DQN is inaccurate on tasks with partially observable states. Modifications in DQN can help with partially observable states

Action-specific Deep Recurrent Q-Network

The goal of this type of DQN with reinforcement learning is to incorporate the influence of the performed action over time. We can achieve this by feeding the performed action and the obtained observation as input to the Q-network. At each time step t the model takes an observation o_t, the action that led to that observation, and the hidden state h_{t-1} from the last forward pass of the LSTM layer. The observation gets fed through a series of convolutional downsampling layers, while the action a_{t-1} gets embedded in a higher dimensional space through a fully connected linear layer (IP in the figure). The downsampled observation and embedded action then get concatenated and fed into an LSTM, which updates its hidden state h_t and outputs a sequence that gets fed into another linearly connected layer. The output size of the final linear layer matches the number of actions to approximate Q values for each possible action. To address the imbalance between the number of actions and the dimensionality of the state representation, a fully connected layer is proposed to be used to embed the one-hot action vectors into a higher dimensional vector. The ADRQN modifies the transition (s_t, a_t, r_t, s_{t+1}) in the experience replay mechanism to $\langle \{a_{t-1}, o_t\}, a_t, r_t, o_{t+1} \rangle$. This allows the framework to fetch the action-observation pair more conveniently [42]. The LSTM layer requires a sequence of action-observation pairs as input during the decision process for a given frame within training or the updating process of the neural network. Therefore, they save the transitions in a sequence $\langle \{a_{t-1}, o_t\}, a_t, r_t, o_{t+1} \}$ within every episode in the replay memory. An effective technique for achieving a precise estimation of the current state in a model-free POMDP problem involves integrating the entire transition history of each episode. However, in the case of Atari games, this can result in thousands of transitions, leading to a significant increase in training cost due to the requirement of unrolling the LSTM layer for a large number of time steps. Here, the LSTM layer is unrolled for 10-time steps during training. In the algorithm of training, we should initialize the parameters of the Q-Network and the Target network with θ and θ^-, respectively. For each episode, the initial action selected is set to "no operation," and the input to the first hidden layer is initialized as a zero

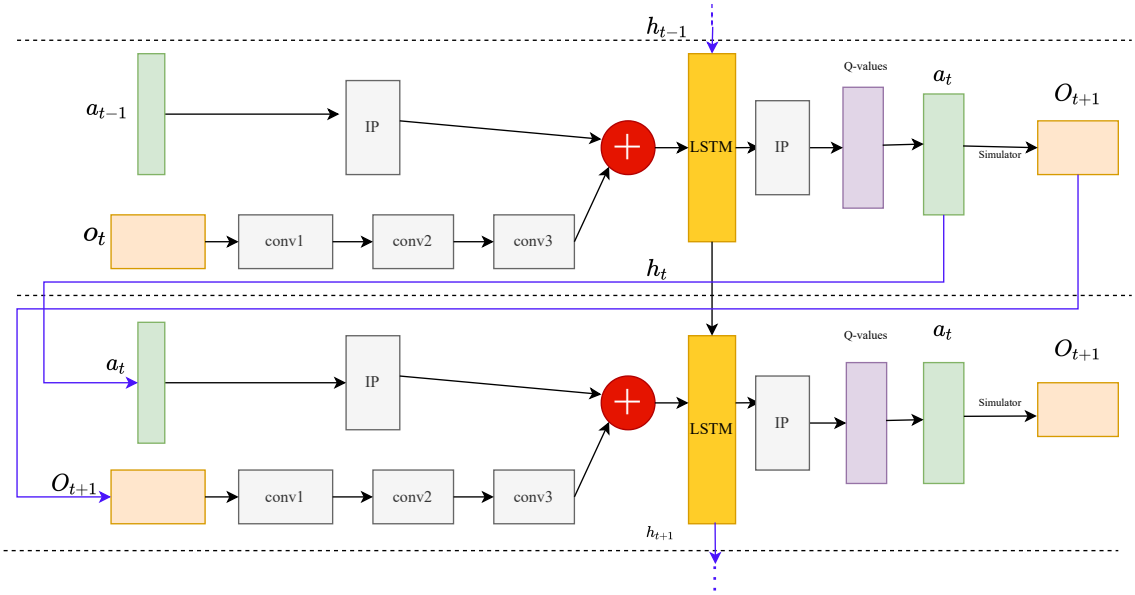

Figure 2.1: ADRQN Network Architecture

vector. Then, the first observation of each episode is initialized with the preprocessed dynamic access. At each time step, the ADRQN select an action based on the epsilon-greedy strategy and execute the action. Next, the agent receives the immediate reward and the subsequent observation of the environment, and the resulting transition is recorded in the current episode's history. The Q-network is updated by randomly selecting and sampling a sequence of transitions $\langle a_{j-1}, o_j, a_j, r_j, o_{j+1} \rangle$ in order to fit the unrolled LSTM layer, and the hidden layers h_{j-1}, h_j of the Q-network and the target network. The difference between these two networks' Q-values Q-values (i.e., Q-value y_j and Q-network value $Q\left(h_{j-1}, a_{j-1}, o_j, a_j; \theta\right)\Big)$ is used as the loss function to update the network parameters via backpropagation [42]. To illustrate how the ADQRN works, we can take a $n_{channels}$ sized vector as input action and pass it through a fully connected layer that outputs 512-D values. The observation is obtained by convolving a single 84 $\times$ 84 frame through 3 convolutional layers, as illustrated in the Figure 2.1. Specifically, the convolutional layers employ 32 8×8 filters with stride 4 ($conv_1$), 64 4×4 filters with stride 2($conv_2$), and 64 3×3 filters with stride 1($conv_3$). The LSTM's 512-D outputs are then fed into another fully connected layer that produces 18-Dimensional Q-values, which correspond to the $n_{actions}$ in the DSA environment.

Chapter 3

State of the Art

Deep reinforcement learning, as a new emerging technology, has significant advantages in environmental automatic exploration and self-decision, and also has great potential in solving dynamic resource allocation problems [43].

Authors in [44] proposed the use of deep Q-learning for spectrum access and propose two methods, DEcentralized spectrum allocation (DESA) and centralized spectrum allocation (CSA), for maximizing network utility.DESA adopts a non-cooperative approach in spectrum decisions while CSA utilizes actions generated through centralized deep Q-network (DQN). Extensive simulations are used to investigate the effectiveness of these proposed schemes for varying primary and secondary network sizes, demonstrating that they outperform model-based RL and traditional approaches, achieving 87% of optimal channel access.

In [45] the authors proposed, two DRL approaches: The deep Q-Network (DQN) and the double deep Q-network (DDQN). Additionally, techniques such as an eligibility trace, prior experience and the "guess process" are introduced to improve DQNs. These approaches were tested in a Markov Decision Process with unknown system dynamics where multiple discrete channels are shared by different types of nodes that lack communication abilities and do not have a priori knowledge of other node behaviors.

In [46] the authors focused on enhancing the spectrum utilization in IoT applications using a DSA scheme based on deep reinforcement learning. The approach involves inter-node collaborations in a dynamic spectrum environment to achieve better access accuracy and efficiency. The proposed scheme incorporates deep double Q-learning for local self-

spectrum-learning, federated learning in edge nodes to improve self-learning, and clustering of multiple secondary users for federated learning to enhance the efficiency of deep reinforcement learning. This proposed scheme, called federated deep reinforcement learning (FDRL), uses global optimization for federated learning, which leads to faster access convergence speed compared to traditional distributed DSA with deep learning. FDRL also preserves the privacy of IoT users by only requiring model parameters to be uploaded to edge servers. The effectiveness of the proposed FDRL-based DSA framework is demonstrated through simulations. J.-M. Kang [47] presents a reinforcement Q-learning algorithm for resource allocation, aimed at minimizing the outage probability of information via channel resource allocation. The proposed method is shown to be highly effective and achieves superior performance, while also satisfying the average power constraint at the energy harvesting node. In [48] proposes a distributed deep reinforcement learning (DRL) based scheme for dynamic spectrum access in multi user wireless networks. Each user selects a channel to transmit packets in a specific time slot with a certain transmission probability. The objective of the proposed scheme is to maximize the network utility by designing a blended strategy for effective DSA. However, in wireless networks, the feedback (ACK packet) may be lost or corrupted due to noise, which presents a challenge in ensuring that multiple agents cooperate to make coherent decisions. To address this challenge, the proposed scheme uses a Deep Recurrent Reinforcement learning network with an integrated GRU layer to optimize the network utility function and a feedback recovery mechanism using complete and incomplete replay buffers. Extensive simulations show the success of the proposed scheme in complex multiuser scenarios and its robustness against the detrimental effects of imperfect feedback.

Yang. K, *et al.* [49] presented a comparative study of three popular deep reinforcement learning algorithms, which are directly applied to wireless network optimization. Both centralized (single agent) and distributed (multi-agent) scenarios were considered. The results showed promising results from both deep deterministic policy gradient and variance based control but also demonstrated certain fragility in neural episodic control (NEC) when the action space is severely limited, this needs further investigation. Muhammad Alrabeiah's [50] work focuses on solving the problem of channel mapping in the space and frequency domains for massive MIMO systems. The proposed solution involves utilizing

a supervised deep learning approach, which effectively reduces the overhead required for both training and feedback.

In order to tackle spectrum allocation per network slice, Yuxiu Hua [51] developed a generative adversarial network-based deep distributional Q-network (GAN-DDQN) using a deep Q-learning approach. The simulation results indicate that this method outperforms traditional deep Q-learning algorithms in terms of performance accuracy.

The authors N Zhang Li *et al.* [52] present a multi-agent deep reinforcement learning model, called neighbor-agent actor-critic (NAAC), designed for spectrum allocation in 6G network device-to-device (D2D) scenarios. The proposed approach employs centralized training, utilizing information from the user's neighbors, and encourages cooperation between users to optimize system performance. The simulation results demonstrate that the NAAC model can enhance the sum rate of D2D links and exhibits excellent convergence. W. Ning *et al.* [53] have developed a Q-learning-based algorithm for channel selection that scans the channel order to minimize overhead and potential delays. This approach achieves higher accuracy and detection probability while also reducing scanning overhead and access delay, compared to state-of-the-art algorithms, leading to enhanced spectrum sharing.

In [54], the authors suggest using a dueling deep recurrent Q-network (Dueling DRQN) based deep reinforcement learning algorithm for dynamic multichannel access in heterogeneous wireless networks. The proposed algorithm aims to achieve high throughput by learning a channel access strategy that makes use of underutilized channels, without prior knowledge of the spectrum environment or other nodes' behaviors. Two key challenges are addressed: the lack of prior knowledge and the partial observability of the spectrum environment with complex temporal dynamics. To address the challenges posed by the dynamic environment, the algorithm utilizes a LSTM layer to consolidate past observations and capture temporal features. In addition, it employs a dueling architecture to overcome the observability problem. Simulation results demonstrate the effectiveness of the proposed approach in various heterogeneous networks, outperforming state-of-the-art policies.

Table 3.1: Summary of related works

Cite	Key Points	Description
[44]	Proposed a decentralized and centralized channel selection method for network utility maximization.	Introduces the deep Q-learning originated spectrum access (DQLS) based DESA and CSA methods.
[45]	Proposed two DRL approaches, DQN and DDQN, and tested them in a Markov Decision Process with unknown system dynamics.	Introduces techniques such as an eligibility trace, prior experience, and the guess process to improve DQNs.
[46]	Proposed the federated deep reinforcement learning (FDRL) based DSA scheme for enhancing the spectrum utilization in IoT applications.	Incorporates deep double Q-learning for local self-spectrum learning, federated learning in edge nodes, and clustering of multiple secondary users for federated learning.
[48]	Proposed a distributed DRL-based scheme for DSA in multiuser wireless networks.	Uses a deep recurrent reinforcement learning network with an integrated GRU layer to optimize the network utility function and a feedback recovery mechanism using complete and incomplete replay buffers.

Table 3.1 continued from previous page

Cite	Key Points	Description
[49]	Presented a comparative study of three popular deep reinforcement learning algorithms applied to wireless network optimization.	Compares deep deterministic policy gradient, variance based control, and neural episodic control in centralized and distributed scenarios.
[50]	Used a novel supervised deep learning approach to reduce overhead in both training and feedback aspects in massive MIMO.	Addressed the issue of channel mapping in space and frequency domain in massive MIMO.
[52]	Proposed the neighbor-agent actor-critic (NAAC) model for spectrum allocation in 6G network D2D scenarios.	Utilizes cooperation between users to optimize system performance, and information from the user's neighbors for centralized training.
[51]	Introduces a deep distributional Q-network (DDQN) powered by a generative adversarial network (GAN), named GAN-DDQN, for spectrum allocation per network slice.	Demonstrated enhanced performance accuracy compared with conventional deep Q-learning algorithms through simulations.

Table 3.1 continued from previous page

Cite	Key Points	Description
[47]	Proposed a reinforcement Q-learning-based algorithm for resource allocation, minimizing the outage probability of information by assigning channel resources.	Demonstrated superior performance and effectiveness of the proposed scheme while satisfying the average power constraint at the energy harvesting node.
[53]	Proposed a Q-learning-based algorithm for channel selection, scanning the order of the channel to reduce overhead and possible delays.	Achieved higher detection probability and accuracy, reduced scanning overhead and access delay compared with the state-of-the-art algorithm through simulations.
[54]	Proposed a deep RL algorithm based on a dueling deep recurrent-Q-network (Dueling DR-QN)	A novel approach that combines the dueling and recurrent techniques to improve RL performance.

Chapter 4

Methodology

4.1 Phases of Problem-Solving

4.1.1 Description of the Problem

The problem with resource allocation is that all users can transmit over the entire spectrum at their assigned power levels and frequency. Thus, creating interference among secondary users as well as among the secondary and primary users. Assuming that the primary user can tolerate interference from the secondary users as long as the total transmission power of all secondary users are below a threshold, the primary user's problem is to design an allocation and a payment mechanism to maximize its expected revenue. Dynamic Spectrum access could be implemented with deep reinforcement learning, where spectrum can be dynamically provisioned to secondary users to maximize long-term rewards for the network and avoid myopic (short-term) decision-making.

4.1.2 Analysis of the Problem

DSA promisingly approach to efficiently allocate radio spectrum to users in a dynamic wireless communication environment. However, traditional spectrum allocation schemes can be inefficient and fail to adapt to changing conditions. DQN, a form of reinforcement learning, has been proposed as a potential solution for DSA. DQN has the ability to learn to select the best spectrum channels based on the current state of the wireless network and the expected reward for selecting a particular channel. In this study, we explore the potential of DQN for DSA and evaluate its performance in a simulated wireless network

environment. Specifically, we investigate the ability of DQN to adapt to changing network conditions and improve spectrum efficiency while reducing interference between wireless devices. The results of this study have important implications for the design of efficient and adaptive wireless communication systems

4.1.3 Algorithm Design

We formulate the problem of dynamic spectrum access as a deep reinforcement learning problem. In this context, we need to define an agent, a state, an action, a reward, and a policy in a dynamic spectrum access environment.

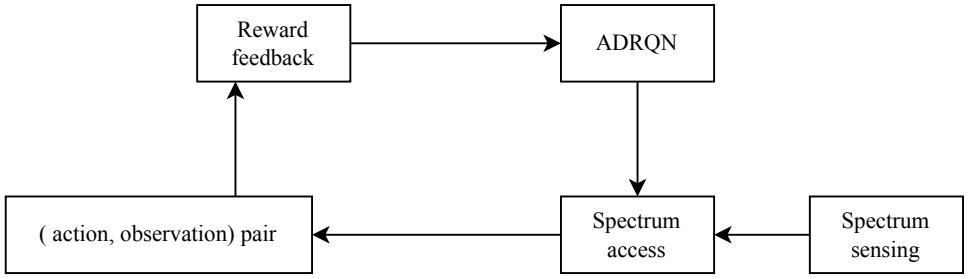

Figure 4.1: Learning Procedure made by a SU with an ADRQN

To transmit data, SUs use spectrum access strategies to find an available sub-band in the wireless channels. They do this by conducting spectrum sensing. After finding an available sub-band, the SU transmitter stores an action and observation pair. It receives feedback rewards based on the actual wireless transmission quality, which is also stored in memory. This memory is then sampled by batches and used to update the spectrum access strategy through training the ADRQN. The learning process must be carried out periodically to adapt to any changes in the wireless environment. We show the learning procedure in Figure 4.1. We can observe that spectrum access strategies are determined by the training of the ADRQN and the actual spectrum sensing.

State

We consider a set of orthogonal channels $1, 2, \ldots, N$ and a set of SUs $1, 2, \ldots, L$. Each channel is occupied by a PU that can be either active (0) or inactive (1). An inactive PU allows an SU to access the corresponding channel, while an active PU denies the SU access.

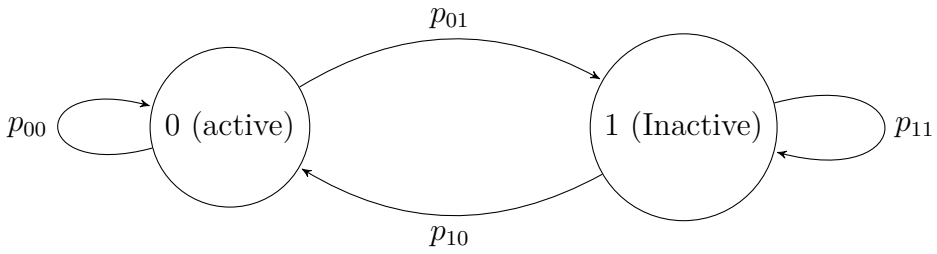

Figure 4.2: Markov Chain that describes the dynamic of PU activity

The activity of each PU is modeled as a two-state Markov chain, shown in Fig. 4.2, with transition probabilities denoted by:

$$P_n = \begin{bmatrix} p_{00}^n & p_{01}^n \\ p_{10}^n & p_{11}^n \end{bmatrix} \tag{4.1}$$

where p_{ij} represents the probability of transitioning to state j given the current state is i $(i, j \in 0, 1)$.

At the start of each time slot, all SUs perform spectrum sensing on all N channels to detect the channel states. The sensing results at time slot t are denoted by

$$S(t) = \left[S^1(t), \ldots, S^L(t)\right] \tag{4.2}$$

where $S^L(t)$ is an N-dimensional vector $\left[s_1^l(t), \ldots, s_N^I(t)\right]^T$, and $s_n^l(t) \in 0, 1$ represents the sensed state of the l-th SU on the n-th channel. Due to imperfect sensing, $s_n^l(t)$ may contain errors with a probability of E_n^l, where $T_n(t)$ denotes the true state of the n-th channel. Both the transition probabilities and sensing error probabilities are unknown to the SUs. The only known information for the l-th SU is $S^l(t)$, which represents the observed state in the environment and the input to the ADRDQN. The architecture of ADRQN is shown in Fig. 4.3.

Actions

After sensing the state of each channel, each SU can decide to either access a channel or remain idle based on the sensing result. The decision of the lth SU is denoted by the variable

$$a^l(t) \in \{0, \ldots, N\} \tag{4.3}$$

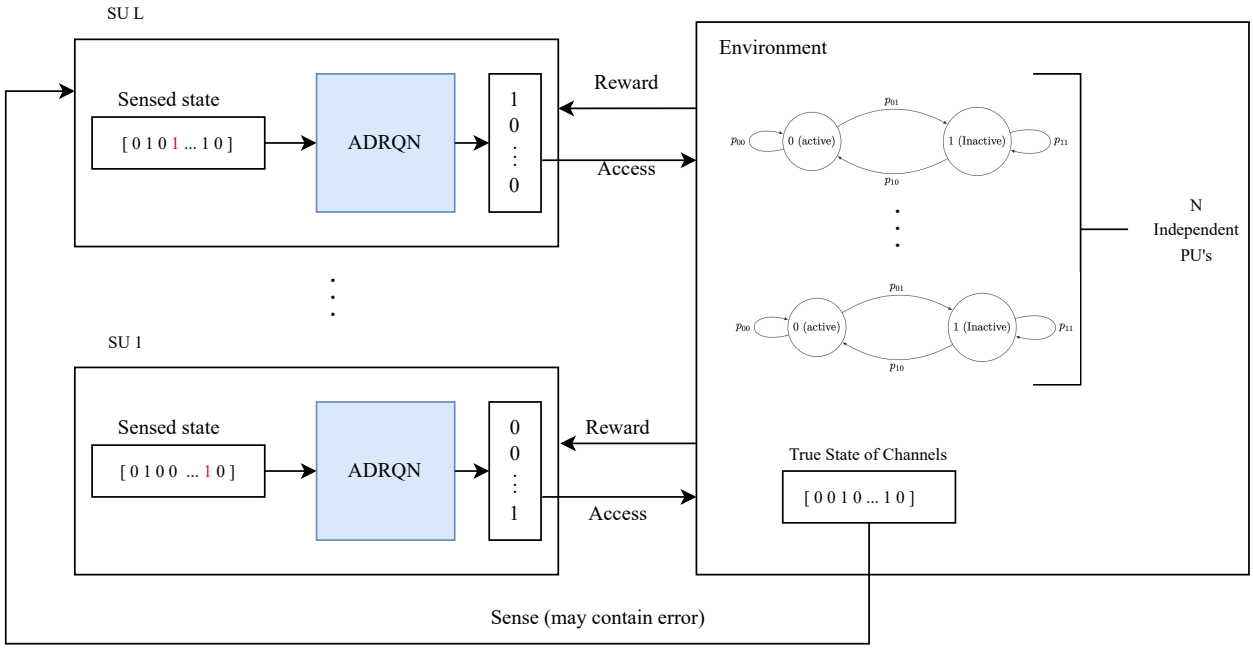

Figure 4.3: ADRQN strategy access

If $a^l(t) = n$ where $n > 0$, it means that the lth SU chooses to access the nth channel at time slot t. On the other hand, if $a^l(t) = 0$, it means that the lth SU chooses not to access any channel at time slot t. When a SU transmitter accesses a channel, the channel's state changes according to its corresponding Markov chain. Then, the corresponding SU receiver provides feedback the received SINR to the transmitter.

Reward

The interference between PUs and SUs will determine the reward function for each SU. There are four cases for setting up the reward function.

- A secondary user (SU) gains access to a channel that is not being used by any primary user (PU) or other secondary user (SU). That is $\log_2(1 + SINR)$, because it does experience no interference.
- If a SU accesses a channel that a PU is using, a negative reward of $-C(C > 0)$ is given to the SU because a collision with the PU occurs. Therefore, the PU will broadcast warning signals to all SUs. Next, the corresponding transmitting SUs will be aware of this collision.

- If more than two SUs access the available channel, collisions happen, and the achievable data transmission rate $\log_2(1 + SINR)$ is used as the reward function
- If a SU decides not to access any channel, the reward is set to zero

Thus, the reward function of the lth SU on the nth channel goes as follows:

$$r^l(t+1) = \begin{cases} -C, & \text{Interference with PU} \\ \log_2\left(1 + SINR_n^l\right), & \text{otherwise} \end{cases} \tag{4.4}$$

Each SU has its DQN to make channel access decisions independently, and the only input to each SU's DQN is the sensing results from its own sensor. Although secondary users (SUs) are unaware of the transition probabilities of channel states and the probabilities of sensing errors, they can acquire knowledge on how to access the channel by monitoring the received SINR. Each SU's goal is to maximize its own cumulative discounted reward, which is defined by [55]:

$$R^l = \sum_{t=1}^{\infty} \gamma^{t-1} r^l(t+1) \tag{4.5}$$

This is the sum of rewards obtained in each time step, discounted by a factor $\gamma \in [0, 1]$. The training method for all DQNs is shown in Algorithm 1.

The architecture of the proposed distributive dynamic spectrum access using an action-specific deep recurrent Q-Network for DSA is shown in Fig. 4.3.

4.1.4 Implementation

The tensorFlow 2.0 API will be used to train the neural network. We need to setup random channel allocation at first place. Then, we have to collect action and observation pairs which are necessary for the neural network. The implementation is based on random data collected in a local machine. Next, the ADRQN is modelled and trained in this local machine.

4.1.5 Testing and Evaluation

For testing and evaluation of the network, we conduct a series of experiments in different scenarios to evaluate how the agents behave and how the network responds to changes in

Algorithm 1 Action-specific Deep Recurrent Q-Network for dynamic spectrum access

Initialize the number of iterations M, the replay memory D.
Initialize Q-Network and Target Network with θ and θ^{-} accordingly
for episode $\leftarrow$ 1 to M **do**
    Initialize the first action $a_0 \leftarrow [0, 0, ...0]$
    Initialize the first observation with the prepossessed first access
    while $o_t \neq$ terminal **do**

$$a_t \leftarrow \begin{cases} \text{random action} & , \text{ w.p. } \epsilon \\ arg\max_a Q(h_{t-1}, a_{t-1}, o_t, a; \theta) & , \text{ otherwise} \end{cases}$$

        Perform action a_t
        Receive r_t, s_{t+1}
        Store transition $\langle \{ a_{t-1}, o_t \}, a_t, r_t, o_{t+1} \rangle$ as one record of the current episode in D
        Randomly sample a mini-batch of transition sequences $a_{j-1}, o_j, a_j, r_j, o_{j+1}$ from D
        Compute Q-value of target network

$$\text{set } y_t \leftarrow \begin{cases} \mathrm{r}_t, & \text{for terminal } \mathrm{s}_{t+1} \\ \mathrm{r}_t + \gamma \max \tilde{Q}_{t+1}, & \text{otherwise} \end{cases}$$

        Compute the gradient of (y_j - $Q(h_{j-1}, a_{j-1}, o_j, a_j; \theta))^2$ ▷ MSE loss
    end while
    Update θ
    if episode % $T_{update} == 0$ **then**
        $\theta^{-} \leftarrow \theta$ ▷ Update target network
end for
return Q**-values**

state. These scenarios include variations in the number of agents, network geometries, and primary users quantity. By doing so, we can gather insights into the network's performance under different conditions and determine how well it generalizes to new situations.

Complexity analysis

Additionally, we perform a rigorous analysis of the complexity of both ADRQN and Q-learning algorithms in order to identify areas for improvement and ensure the network's robustness and scalability.

- **Complexity analysis of Q-learning algorithm** The Q-learning algorithm has a time complexity of $O(MT)$. It becomes stable and converges after T time slots and M rounds of iteration. Assuming that each iteration takes one unit of time,

the optimal allocation strategy obtained through multiple iterations also has a time complexity of $O(MT)$. Therefore, the Q-learning algorithm's time complexity is also $O(MT)$.

- **Complexity analysis of DQN algorithm**

  The ADRQN algorithm has a time complexity of $O\left(MT\sum_{l=0}^{L-1} u_l u_{l+1}\right)$. This means that after T time slots and M rounds of iteration, the neural network parameters of the DQN algorithm will stabilize and become consistent. The time complexity of a neural network is determined by the input state and action dimensions, as well as the number and layers of neurons in each layer. The number of layers of DQN neural network is represented by L, and the number of neurons in layer l is represented by u_l.

4.2 Model Proposal

4.2.1 Environment

We examine a dynamic spectrum access (DSA) scenario where N Primary Users (PUs) operate alongside L Secondary Users (SUs). The number of wireless channels matches the number of PUs, so each PU transmits on a unique channel to avoid interference. If a collision occurs, each PU must broadcast warning signals to SUs to protect itself. This warning signals are implemented to protect the authorized PU from harmful interference by SUs. Our proposed approach aims to surpass the myopic method, which requires knowledge of system statistics, and converge faster than an Echo State Network that employs frozen neurons in the feed-forward layer. We present benchmarks with the Q-learning method, which also converges faster when the number of channels is large. Each SU has access to all wireless channels and can select the appropriate channel with its spectrum access strategy.

We denote the transmitter and receiver coordinates of SU i, the transmitter and receiver coordinates of PU j as $\left(x_i^{TX}, y_i^{TX}\right)$, $\left(x_i^{RX}, y_i^{RX}\right)$, $\left(x_j^{TX}, y_j^{TX}\right)$, and $\left(x_j^{RX}, y_j^{RX}\right)$, respectively. The Euclidean distance of a desired signal link is computed as

$$d_{ii} = \sqrt{\left(x_i^{TX} - x_i^{RX}\right)^2 + \left(y_i^{TX} - y_i^{RX}\right)^2} \tag{4.6}$$

The propagation distance of interference signals from other SUs is given by

$$d_{ji} = \sqrt{\left(x_j^{TX} - x_i^{RX}\right)^2 + \left(y_j^{TX} - y_i^{RX}\right)^2} \tag{4.7}$$

and

$$d_{ki} = \sqrt{\left(x_k^{TX} - x_i^{RX}\right)^2 + \left(y_k^{TX} - y_i^{RX}\right)^2} \tag{4.8}$$

where $k \in 1, 2, \ldots, L$ and $k \neq i$. Interference only occurs when PU j and SU k use the same wireless channel as SU i.

$$PL\left(d, f_c\right) = \overline{PL} + A_W \cdot \log_{10}(d[m]) + B_W \cdot \log_{10}\left(\frac{f_c[GHz]}{5}\right) \tag{4.9}$$

Here, f_c denotes the carrier frequency of wireless channels, $\overline{PL}$ represents the path loss at a reference distance, A_W denotes the path loss exponent, and B_W denotes the path loss frequency dependence. The path loss of the desired signal $PL\left(d_{ji}, f_c\right)$ and interference signals $PL\left(d_{ji}, f_c\right)$ and $PL\left(d_{ki}, f_c\right)$ can be determined. We assume a strong Line-of-Sight (LOS) path between a transmitter and a receiver, allowing us to employ the Rician channel model to derive the channel model, which can be expressed as:

$$h = \sqrt{\frac{\kappa}{\kappa + 1}}\sigma e^{j\theta} + \sqrt{\frac{1}{\kappa + 1}} CN\left(0, \sigma^2\right) \tag{4.10}$$

The variable κ is a parameter that indicates the ratio of the power of the line-of-sight (LOS) path to the power of the scattered paths. The variable $\theta \sim U(0, 1)$ is the phase of the arrival signals on the LOS path and is randomly selected from a uniform distribution between 0 and 1. The notation $CN(\cdot)$ denotes a circularly symmetric complex Gaussian random variable. Meanwhile,

$$\sigma^2 = 10^{-\frac{PL(d, f_c)}{10}} \tag{4.11}$$

The variable σ^2 is determined and calculated based on the path loss and other parameters. It is determined by using a formula that takes into account the average path loss, the distance between the transmitter and the receiver, the carrier frequency, and other parameters.

Finally, the SINR of the received signals at the receiver of SU i can be obtained as

follows:

$$SINR_i = \frac{p_{ij} \cdot |h_{ii}|^2}{p_{jj} \cdot \left|h_{ji}\right|^2 + \sum_{k=1,k\neq i}^{L} p_{kj} \cdot |h_{ki}|^2 + B \cdot N_0} \tag{4.12}$$

The transmit power of PU j, SU i, and SU k on the j-th wireless channel, are represented by the variables p_{jj}, p_{ij}, and p_{kj}, respectively. The terms $|h_{ii}|^2$, $\left|h_{ji}\right|^2$, and $|h_{ki}|^2$ represent the channel gains of the links between transmitter i and receiver i, transmitter j and receiver i, and transmitter k and receiver i, respectively. The variables B and N_0 stand for channel bandwidth and noise spectral density, respectively. Given the received SINR of SU i, we can calculate the channel capacity using $C_i = B \cdot \log_2(1 + \text{SINR}_i)$.

4.3 Experimental Setup

In a 150m x 150m square, we select the locations of SUs and PUs randomly. The distance between a SU's transmitter and its corresponding receiver is chosen randomly from a range of 20m to 40m. We apply the WINNER II and Rician models to compute the path loss and our custom channel model, respectively. Specifically, for the WINNER II model, we assign $f_c = 5$GHz, $\overline{PL} = 41$, $A_W = 22.7$, and $B_W = 20$. Regarding the Rician model, we set κ to 8 and σ^2 based on the path loss obtained from the WINNER II model. We then calculate the received SINR at a SU's receiver, with a bandwidth of $B = 1$MHz, a noise spectral density of $N_0 = 10^{-14.7}$(mW/Hz), a transmit power of 20 mW for one SU, and a transmit power of 40 mW for one PU. Table 4.1 lists all the system parameters used in the channel model.

The PU states are modeled using separate two-state Markov chains, where the states are either inactive (1) or active (0). To setup each Markov chain, we randomly choose p_{11} and p_{00} from a uniform distribution over the ranges of $[0.7, 1]$ and $[0, 0.3]$, respectively, for every channel. We can then calculate $p_{10} = 1 - p_{11}$ and $p_{01} = 1 - p_{00}$. We selected the ranges for p_{11} and p_{00} based on the low utilization of most licensed spectrum bands. Therefore, the possible value of p_{11} should be high, and the possible value of p_{00} should be low.

N_0	$10^{-14.7}(mW/Hz)$
B	$1Mhz$
B_W	20
A_W	22.7
PL	41
f_c	$5GHz$
SU transmit power	$20mW$
PU transmit power	$40mW$

Table 4.1: Channel parameters

Chapter 5

Results and Discussion

This chapter presents the final experimental results made by our deep recurrent Q-network called action-specific deep recurrent Q-network for dynamic spectrum access. The results include the ADRQN analysis in terms of loss. Also, we include results on two scenarios. The first scenario includes six primary users with two secondary users trying two make access. The second scenario relates to 20 secondary users, which is a better real-world problem, and five channels. The primary users in this scenario are occupying the spectrum most of the time.

5.0.1 ARDQN hyperparameters

We describe our deep neural network hyperparameters in Table 5.1. This values are the default in DRL algorithms and work well on our case of use. We set the learning rate to 0.01, and $\gamma = 0.95$. Figure 5.1 shows the DQN loss over the course of training. The x-axis represents the training steps during training, while the y-axis represents the cost or loss value. The loss starts high at the beginning of training with a relative value of 40, indicating that the DQN is making random or poorly informed decisions. However, as training progresses, the loss gradually decreases indicating that the DQN is learning to select better actions and optimize its policy. Eventually, the loss converges to a low value of 2, indicating that the DQN has learned to accurately predict the optimal Q-values for each action. This is a good sign that the DQN has successfully learned to navigate the environment and optimize its performance according to the reward function. Overall, the figure demonstrates that the DQN is effectively learning from the environment and

improving its performance over time.

By training an agent to learn optimal actions based on observed states and rewards, The algorithm help dynamically select appropriate frequency bands, and adjust transmission parameters in real-time. This adaptive spectrum management can lead to improved spectrum utilization, reduced interference, and enhanced overall system performance.

Learning rate (lr)	0.01
Replay buffer	10000
Gamma (γ)	0.95
Epsilon (ϵ)	0.9

Table 5.1: ADRQN Hyperparameters settings

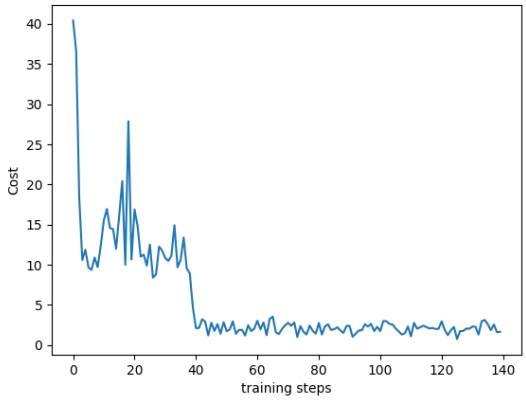

Figure 5.1: Training Loss in ADRQN

5.0.2 Scenario 6 PU, 2 SU

In this section, we design the environment with 6 primary users and 2 secondary users. We can see the real environment geometry shown in Figure 5.2. These locations are randomized.

Figure 5.3 (a) show that there are always collisions when a SU tries to access the channel at some time. Therefore, we calculate an average collision rate and we can see that with our proposed method, there are fewer collisions with our method compared with other, after certain episodes. Comparing with the method that has more collision, we show an improvement of 25% with our method.

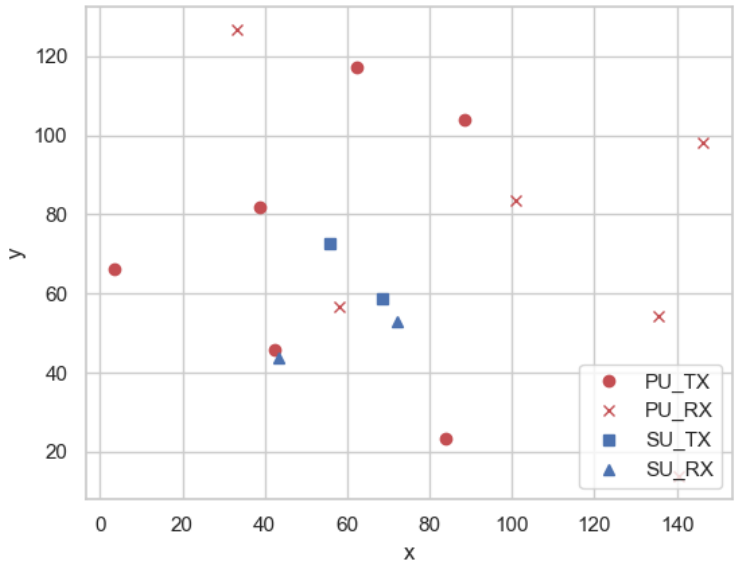

Figure 5.2: Network Geometry with 2 Secondary Users.

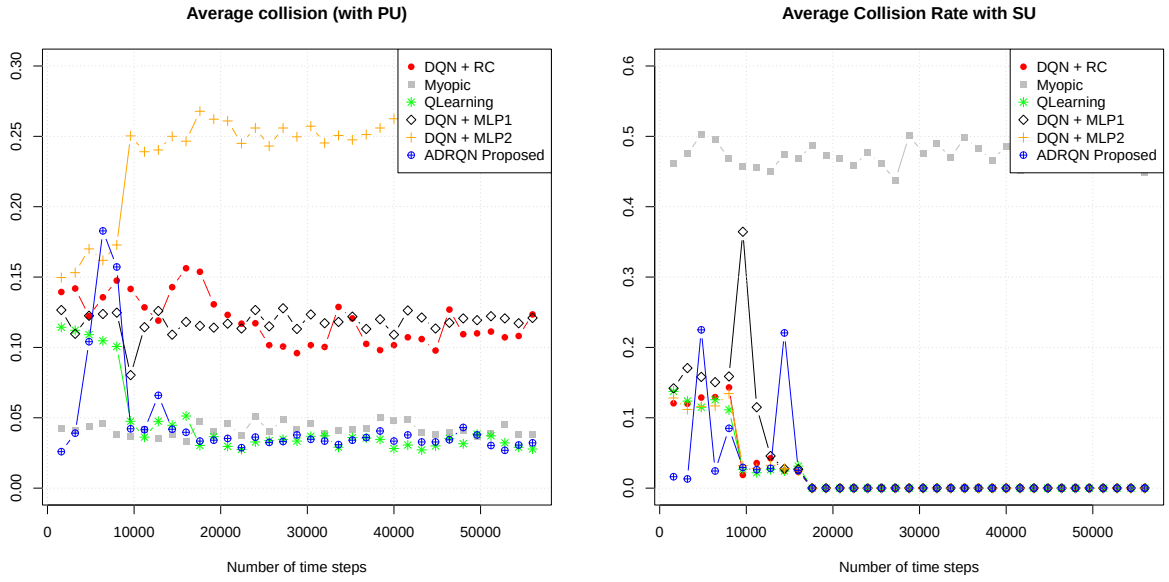

Figure 5.3: **(a)** Average Collision Rate with PUs, scenario 6 PU, 2 SU **(b)** Average Collision Rate with SUs, scenario 6 PU, 2 SU

Figure 5.3 (b) shows that collisions can occur between primary users and secondary users who attempt to access the channel simultaneously. It is observed that methods such as the Myopic method exhibit erratic behavior and lead to collisions among secondary users. However, our method surpasses previous methods by ensuring no collisions among secondary users. It is expected because secondary users has feedback about the environment when they are in the sensing stage. By learning from experience, our algorithm can identify patterns, detect spectrum opportunities, and predict future spectrum availability. This enables more accurate and proactive decision-making, allowing DSA systems to anticipate and exploit spectrum availability efficiently. Consequently, with our neural network, it is expected to overcome the partial observable problem. Figure 5.4 (a) demonstrates that agents receive a reward through reinforcement learning. Our method has been successful in achieving close to the maximum possible reward on average. As the agents learn how to access the communication channels, the SINR has improved, resulting in a good signal and no interference with Primary Users (PU). Figure 5.4 (b) shows that with our method we overcome the other methods like the DQN + RC and a based method like Q-learning. This is because our method has more success rate in accessing channels, and overpasses by the myopic method by a factor of 5.

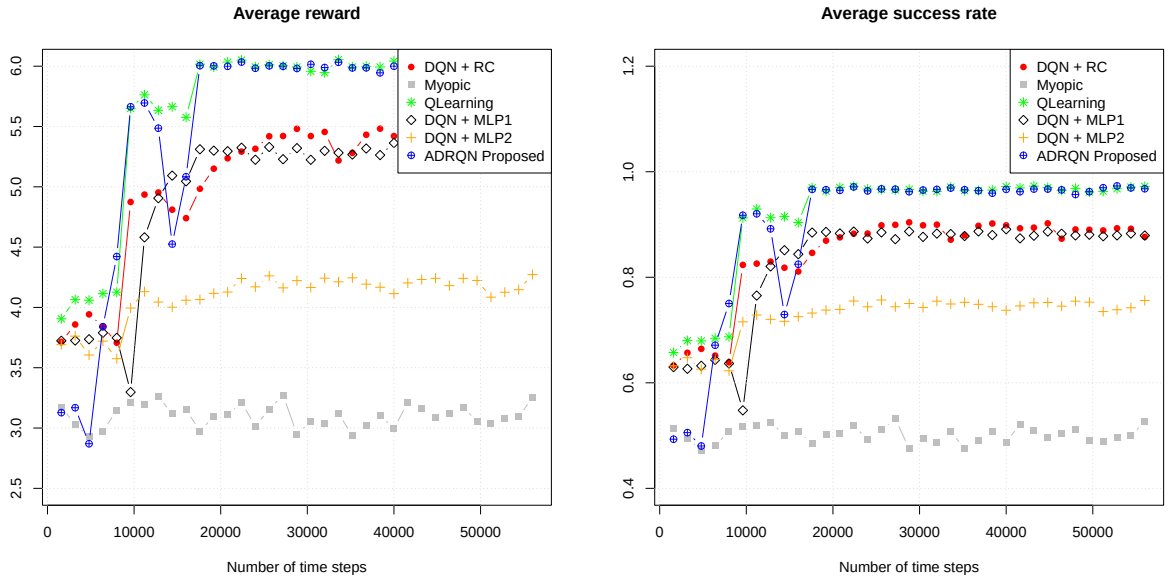

Figure 5.4: **(a)** Average Reward, scenario 6 PU, 2 SU **(b)** Average Success Rate on accessing, scenario 6 PU, 2 SU

5.0.3 Scenario 5 channels and 20 SUs

The scenario with five channels and 20 secondary users performed the best overall. We can see the network geometry on the Figure 5.5.

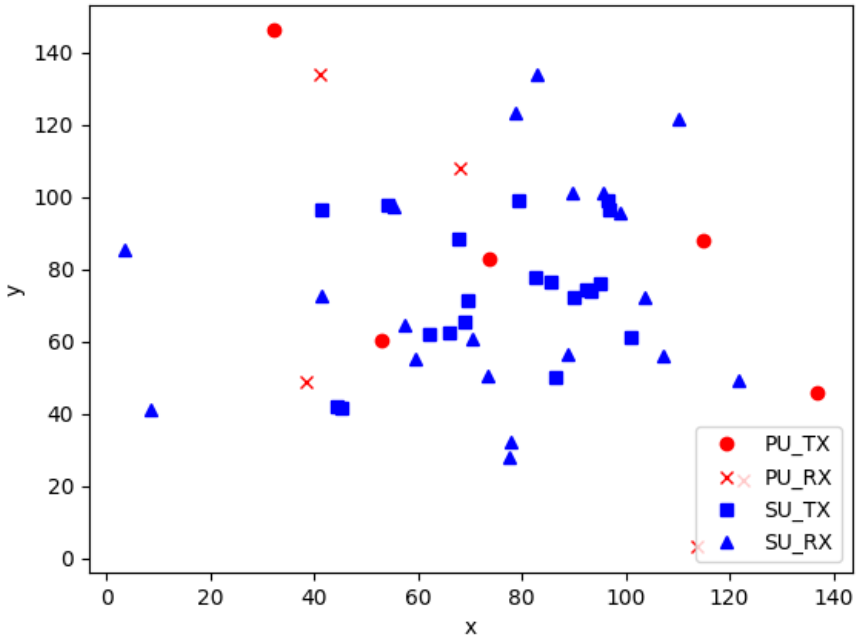

Figure 5.5: Network geometry with 20 secondary users, users are randomly place between 20m and 40m apart

The figure shows that the environment is more crowded, simulating almost a real world scenario. This is useful for comparing our method with a Q-learning approach where it fails when the state is large.

The ADRQN algorithm outperformed all other methods in the neural network study. The graph showed some peaks during the learning process, but after 12000 iterations, the increase became more consistent, indicating that the agents were starting to learn.

Our algorithm leverage the concept of experience replay, which allows agents to learn from past experiences stored in a replay memory. This capability is particularly valuable in DSA, as it enables agents to learn from historical data and adapt to long-term patterns and dynamics in spectrum availability, improving decision-making and performance.

Figure 5.6 demonstrated that when a secondary user successfully accessed a channel, it was a sign of learning. Our network had a higher success rate compared to other methods.

Our algorithm provides a framework for training secondary users that can adapt and make intelligent decisions in dynamic and uncertain spectrum environments. It can handle changing interference patterns and varying channel conditions. By continuously learning from interactions with the environment, our agents can adapt their behavior and strategies to optimize performance under different conditions.

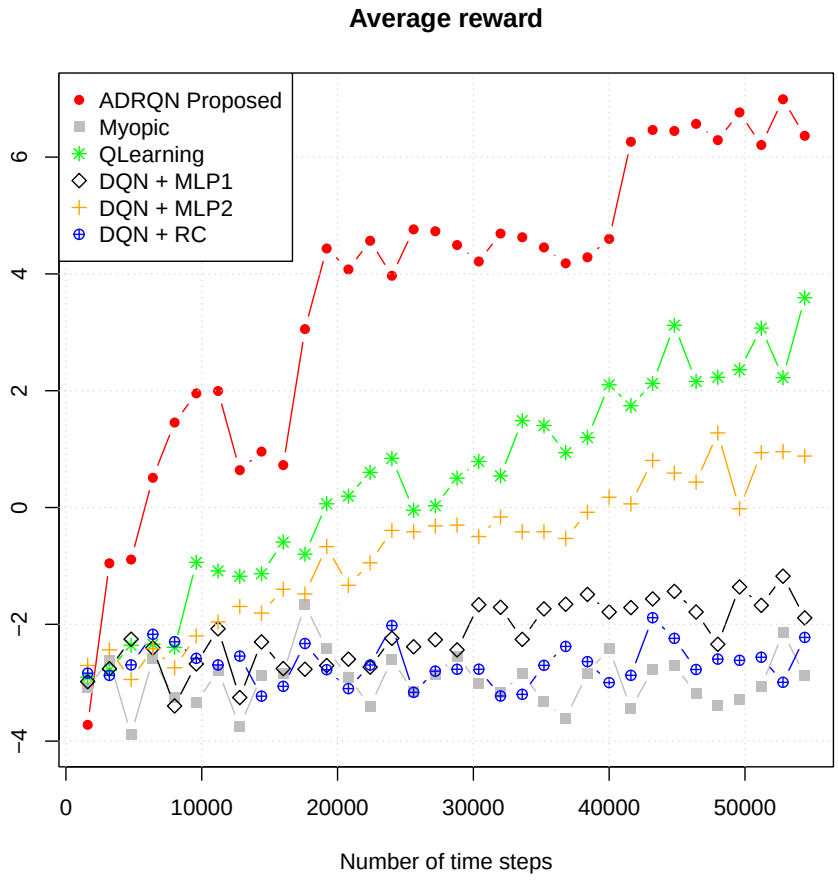

Figure 5.6: Average Reward, scenario 5 channels and 20 SUs

In Figure 5.7, we can observe the success rate comparison between our proposed method and the Q-Learning method. Our approach has a consistently higher success rate, indicating that our method is more effective in optimizing the decision-making process in partial observable domains. The success rate is measured as the percentage of episodes where the secondary user successfully accesses the channel without interfering with the primary user. Our deep neural network's improved training space allows our algorithm to make more accurate decisions, resulting in a higher success rate.

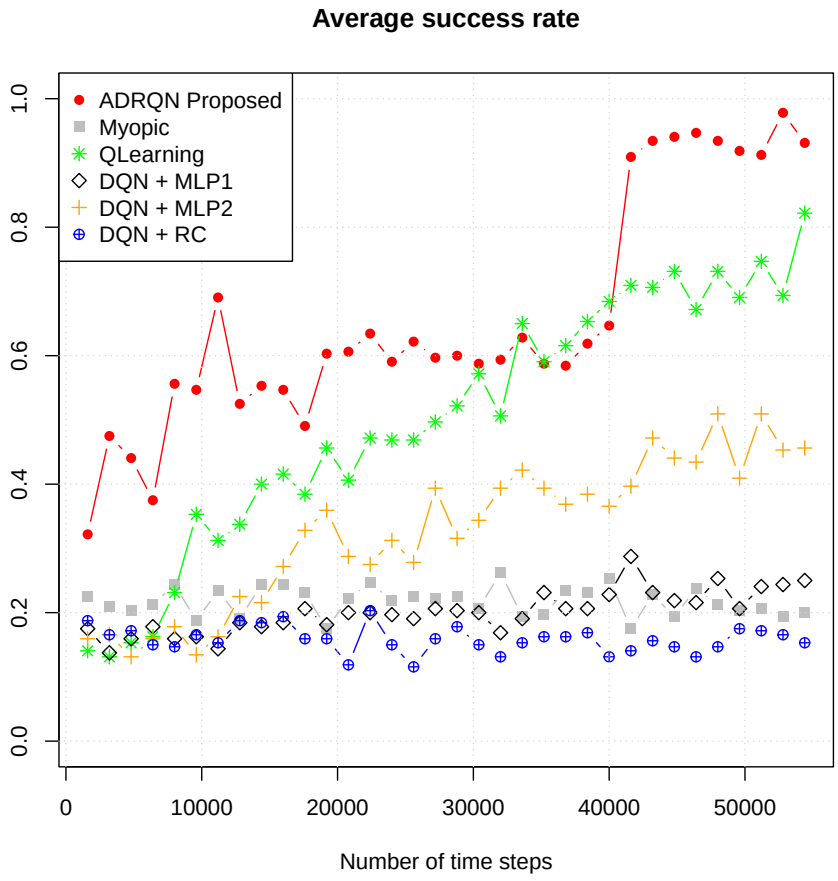

Figure 5.7: Average Success Rate, scenario 5 channels and 20 SUs

Furthermore, Figure 5.8 shows the average collision rate of our proposed method during the learning process. We can see that the average collision rate stabilizes within 12000 iterations, which indicates that our algorithm has effectively learned the optimal channel access strategy. This suggests that our approach is suitable for real-world scenarios, where channel access is dynamic, and the secondary user needs to adapt quickly.

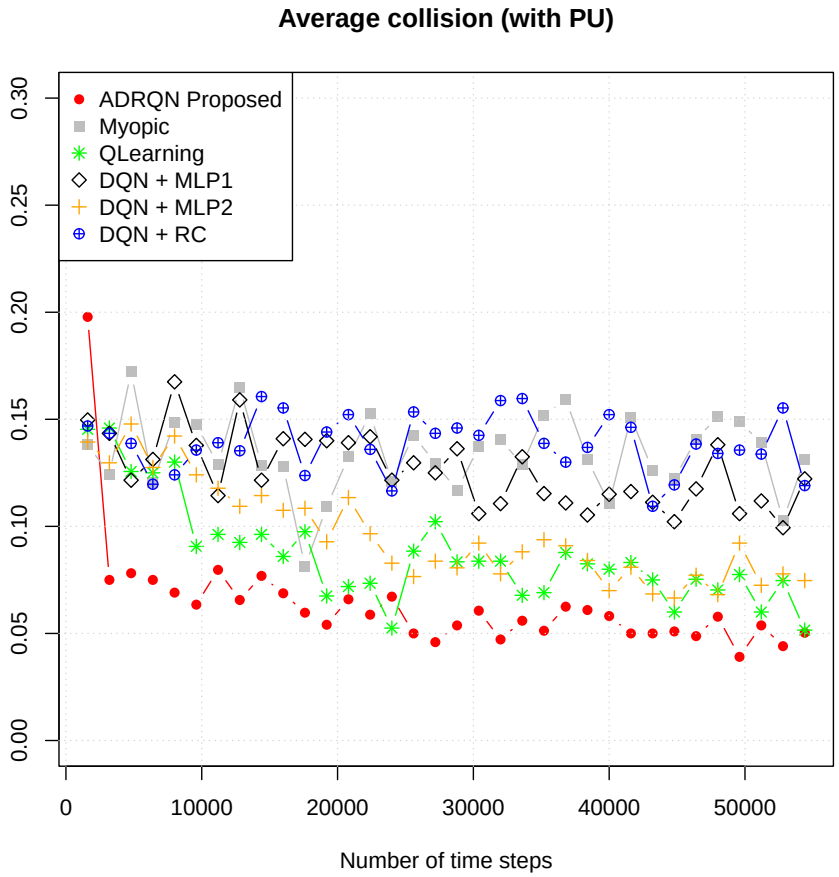

Figure 5.8: Average Collision Rate with PU, scenario 5 channels and 20 SUs

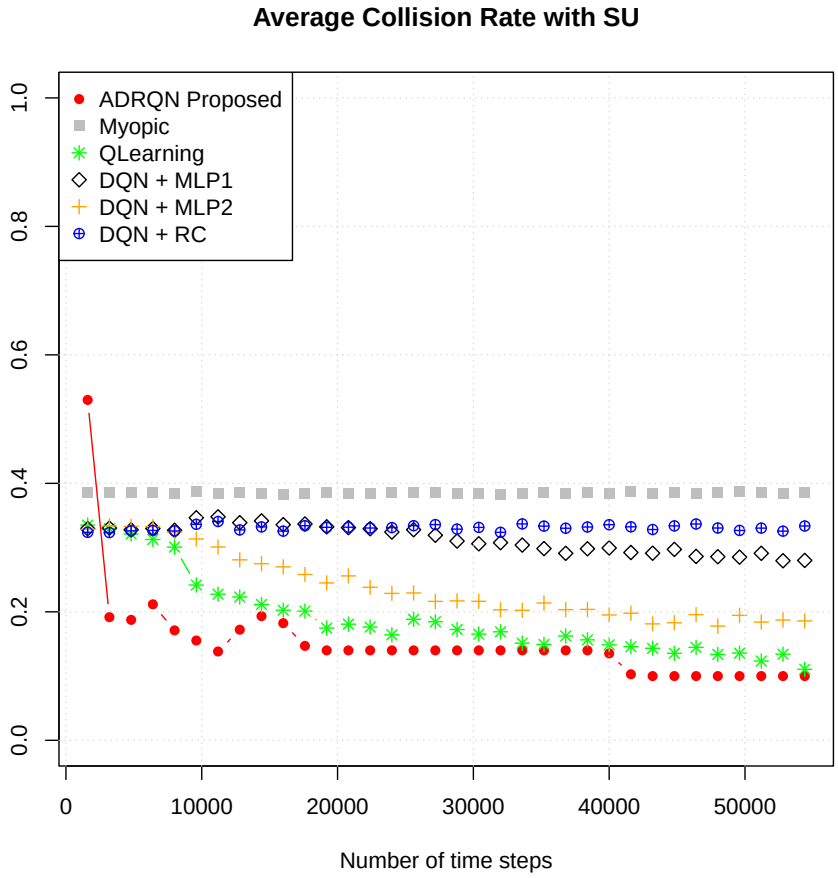

Figure 5.9: Average Collision Rate with SU, scenario 5 channels and 20 SUs

Finally, Figure 5.9 illustrates the absence of collisions between the secondary and primary users after 50000 iterations. This indicates that our proposed approach can effectively reduce the interference between the primary and secondary users, ensuring the stability and reliability of the communication system. The absence of collisions is a critical measure of the success of any channel access method, and our approach demonstrates significant improvements over the Q-Learning method in this aspect.

5.0.4 Complexity analysis

DRL algorithms have a higher time complexity than Q-Learning due to the computational overhead of the neural network training process. However, the advantages of DRL algorithms in handling complex and high dimensional input data often outweigh the computational cost, making them a popular choice in various applications. Using Q-learning

we can define a state-action value function $Q(s, a)$ that estimates the expected future reward for taking action in state s. We update this function using the Bellman equation and use an epsilon-greedy policy to select actions.

On the other hand, deep reinforcement learning uses a deep neural network to approximate the state-action value function. This approach has the advantage of being able to generalize across similar states, reducing the number of updates required. However, training the neural network requires a large amount of data and computing power.

We can see in Figure 5.10 that Q-learning has a relatively constant time complexity as the size of the environment increases. In contrast, deep reinforcement learning exponentially increases in time complexity as the number of parameters in the neural network grows. This makes Q-learning a more suitable approach for simple environments with few states and actions. At the same time, deep reinforcement learning is better suited for complex environments with many states and actions where generalization is required. In our case, as the number of channels increase to 20, the time complexity is twice as Q-learning, but it starts converging to a constant value.

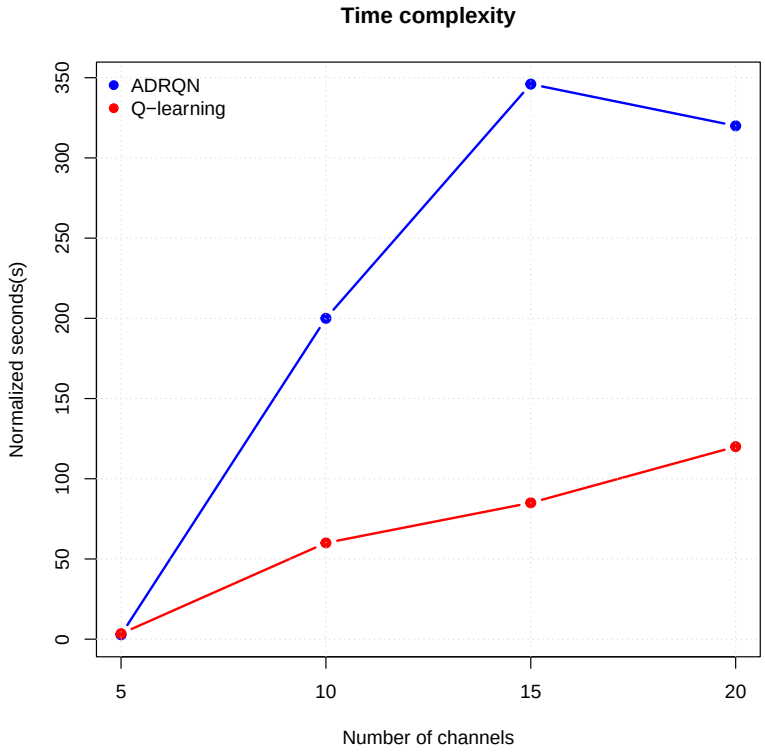

Figure 5.10: Time complexity comparison between Q-learning and ADRQN (ours)

Chapter 6

Conclusions

DSA networks address current problems and issues in wireless networks caused by limited spectrum availability and poor spectrum utilization by using sensed spectrum in a dynamic, clever, and opportunistic manner.

This thesis proposed a new action-specific architecture based on deep reinforcement learning for partially observable environments. Especially our proposed schemes consist of feeding the LSTM layer with a sequence of action-observation pairs. Thus, we store the transitions sequentially in the replay memory that can be unrolled. In our experiments, we unrolled the LSTM layer for 10 time steps during training. Our results show that with this method, we can improve accuracy in channel assignment compared with previous work.

Regarding the results presented in the chapter, it can be concluded that the proposed deep recurrent Q-network, specifically the action-specific deep recurrent Q-network, is an effective method for dynamic spectrum access in scenarios with multiple primary and secondary users. In terms of the ADRQN hyperparameters, the results showed that setting the learning rate to 0.01 and γ to 0.95 was effective in reducing the loss over time, indicating that the network was learning to optimize its policy and make better decisions. In the scenario with six primary users and two secondary users, the proposed method showed a significant improvement in reducing collisions compared to other methods, achieving a 25% improvement over the method with the highest collision rate. The method was also effective in reducing collisions between primary and secondary users, ensuring that secondary users could access the channels without interfering with primary users. In the scenario with five channels and 20 secondary users, the proposed method outperformed

other methods in terms of success rate, collision rate, and interference between primary and secondary users. The method achieved a consistently higher success rate compared to the Q-learning method, indicating that the proposed method was more effective in making accurate decisions in partially observable domains. The method was also able to effectively reduce the collision rate and interference between primary and secondary users, ensuring the stability and reliability of the communication system. Finally, it is important to note that DRL algorithms, including the proposed method, have a higher time complexity than Q-learning due to the computational requirements of training deep neural networks. However, the proposed method demonstrated its effectiveness in dynamic spectrum access, which is a critical application in wireless communication networks.

In conclusion, deep reinforcement learning has shown promising results for dynamic spectrum access in wireless communication systems. DRL approaches can adapt to changing environments and make efficient use of the available spectrum resources. However, DRL methods still face challenges such as scalability and stability, and further research is needed to fully realize their potential in dynamic spectrum access. Additionally, DRL techniques are not without limitations, and the selection of the best DRL algorithm for a given problem is highly dependent on the specific problem and the underlying wireless network characteristics.

For future work, we are planning to implement and improve the performance in more real-world scenarios and use a new model way of dealing with sequential data that is called Transformers and do hyper-parameter tuning.

Bibliography

[1] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[2] M. A. Matin, Ed., *Spectrum Access and Management for Cognitive Radio Networks*. Springer Singapore, 2017. [Online]. Available: https://doi.org/10.1007%2F978-981-10-2254-8

[3] H.-H. Chang, H. Song, Y. Yi, J. Zhang, H. He, and L. Liu, "Distributive dynamic spectrum access through deep reinforcement learning: A reservoir computing-based approach," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1938–1948, 2019.

[4] M. Haenggi, J. G. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti, "Stochastic geometry and random graphs for the analysis and design of wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 7, pp. 1029–1046, 2009.

[5] J. Meinil, P. Kysti, T. Jms, and L. Hentil, "WINNER II channel models," in *Radio Technologies and Concepts for IMT-Advanced*. John Wiley & Sons, Ltd, pp. 39–92. [Online]. Available: https://doi.org/10.1002/9780470748077.ch3

[6] J. M. Chapin and W. H. Lehr, "Cognitive radios for dynamic spectrum access - the path to market success for dynamic spectrum access technology," *IEEE Communications Magazine*, vol. 45, no. 5, pp. 96–103, 2007.

[7] *Cognitive Radio Communications and Networks*. Elsevier, 2010. [Online]. Available: https://doi.org/10.1016/c2009-0-19335-2

[8] I. F. Akyildiz, W.-Y. Lee, and K. R. Chowdhury, "CRAHNs: Cognitive radio ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 810–836, Jul. 2009. [Online]. Available: https://doi.org/10.1016/j.adhoc.2009.01.001

[9] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, Sep. 2006. [Online]. Available: https://doi.org/10.1016/j.comnet.2006.05.001

[10] A. Al-Saman, M. Cheffena, O. Elijah, Y. A. Al-Gumaei, S. K. Abdul Rahim, and T. Al-Hadhrami, "Survey of millimeter-wave propagation measurements and models in indoor environments," *Electronics*, vol. 10, no. 14, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/14/1653

[11] B. J. Wild and K. Ramchandran, "Detecting primary receivers for cognitive radio applications," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.*, pp. 124–130, 2005.

[12] Q. Zhao, "Spectrum opportunity and interference constraint in opportunistic spectrum access," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*. IEEE, Apr. 2007. [Online]. Available: https://doi.org/10.1109/icassp.2007.366752

[13] O. Ileri, D. Samardzija, T. Sizer, and N. Mandayam, "Demand responsive pricing and competitive spectrum allocation via a spectrum server," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.* IEEE. [Online]. Available: https://doi.org/10.1109/dyspan.2005.1542635

[14] K. Hakim, S. K. Jayaweera, G. El-Howayek, and C. Mosquera, "Efficient dynamic spectrum sharing in cognitive radio networks: Centralized dynamic spectrum leasing (c-dsl)," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 2956–2967, 2010.

[15] G. Ganesan and Y. Li, "Cooperative spectrum sensing in cognitive radio networks," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.* IEEE. [Online]. Available: https://doi.org/10.1109/dyspan.2005.1542628

[16] F. Khozeimeh and S. Haykin, "Brain-inspired dynamic spectrum management for cognitive radio ad hoc networks," *IEEE Transactions on Wireless Communications*,

vol. 11, no. 10, pp. 3509–3517, Oct. 2012. [Online]. Available: https://doi.org/10.1109/twc.2012.081312.111538

[17] G. Ganesan and G. Y. Li, "Cooperative spectrum sensing in cognitive radio, part i: Two user networks," *IEEE Transactions on Wireless Communications*, vol. 6, pp. 2204–2213, 2007.

[18] C. Luscht, M. Sandell, P. Strauch, J.-J. Wu, C. Ilas, P.-W. Ong, R. Baeriswyl, F. Battaglia, K. Spyros, and R.-H. Yan, "Advanced signal-processing algorithms for enery-efficient wireless comunications," *Proceedings of the IEEE*, vol. 88, no. 10, pp. 1633 – 1649, 2000. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0001653209&doi=10.1109%2F5.889000&partnerID=40&md5=026b9a0407328a47964c0080c055358a

[19] P. Mangayarkarasi, M. Ramya, and S. Jayashri, "Analysis of various power allocation algorithms for wireless networks," in *2012 International Conference on Communication and Signal Processing*, 2012, pp. 133–136.

[20] U. Phuyal, S. C. Jha, and V. K. Bhargava, "Joint zero-forcing based precoder design for qos-aware power allocation in mimo cooperative cellular network," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 2, pp. 350 – 358, 2012.

[21] D. Wang, Z. Li, and X. Wang, "Joint Optimal Subcarrier and Power Allocation for Wireless Cooperative Networks Over OFDM Fading Channels," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 249–257, 2012.

[22] H.-L. Xiao, P. Wang, S. Ouyang, and M.-Z. Li, "Power allocation scheme based on system capacity maximization for multi-base station cooperative communication," *Beijing Youdian Daxue Xuebao/Journal of Beijing University of Posts and Telecommunications*, vol. 36, pp. 93–97, 2013.

[23] Z. WU and H.-b. YANG, "Power allocation of cooperative amplify-and-forward communications with multiple relays," *The Journal of China Universities of Posts and Telecommunications*, vol. 18, no. 4, pp. 65–69, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1005888510600854

[24] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, Dec. 1943. [Online]. Available: https://doi.org/10.1007/bf02478259

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986. [Online]. Available: https://doi.org/10.1038/323533a0

[26] Y.-S. Park and S. Lek, "Chapter 7 - artificial neural networks: Multilayer perceptron for ecological modeling," in *Ecological Model Types*, ser. Developments in Environmental Modelling, S. E. Jørgensen, Ed. Elsevier, 2016, vol. 28, pp. 123–140. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780444636232000074

[27] B. Yegnanarayana, *Artificial Neural Networks*. Prentice-Hall of India Pvt.Ltd, 2004.

[28] P. Cunningham, M. Cord, and S. J. Delany, "Supervised learning," in *Machine Learning Techniques for Multimedia*. Springer Berlin Heidelberg, pp. 21–49. [Online]. Available: https://doi.org/10.1007/978-3-540-75171-7_2

[29] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *INTERSPEECH*, 2014, pp. 338–342.

[30] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000. [Online]. Available: https://doi.org/10.1162/089976600300015015

[31] "10.1162/153244303768966139," *CrossRef Listing of Deleted DOIs*, vol. 1, 2000. [Online]. Available: https://doi.org/10.1162/153244303768966139

[32] T. G. Dietterich, "Hierarchical reinforcement learning with the maxq value function decomposition," 1999. [Online]. Available: https://arxiv.org/abs/cs/9905014

[33] R. S. Sutton and A. G. Barto, *Reinforcement Learning, second edition: An Introduction*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2018. [Online]. Available: https://books.google.com.ec/books?id=uWV0DwAAQBAJ

[34] A. Graves, A. rahman Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, May 2013. [Online]. Available: https://doi.org/10.1109/icassp.2013.6638947

[35] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 273–278.

[36] N. Jaouedi, N. Boujnah, and M. S. Bouhlel, "A new hybrid deep learning model for human action recognition," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 4, pp. 447–453, May 2020. [Online]. Available: https://doi.org/10.1016/j.jksuci.2019.09.004

[37] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[38] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013. [Online]. Available: https://proceedings.neurips.cc/paper/2013/file/1ff8a7b5dc7a7d1f0ed65aaa29c04b1e-Paper.pdf

[39] H.-n. Wang, N. Liu, Y.-y. Zhang, D.-w. Feng, F. Huang, D.-s. Li, and Y.-m. Zhang, "Deep reinforcement learning: a survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 12, pp. 1726–1744, 2020. [Online]. Available: https://doi.org/10.1631/FITEE.1900533

[40] G. Tesauro, "Practical issues in temporal difference learning," in *Advances in Neural Information Processing Systems*, J. Moody, S. Hanson, and R. Lippmann, Eds.,

vol. 4. Morgan-Kaufmann, 1991. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1991/file/68ce199ec2c5517597ce0a4d89620f55-Paper.pdf

[41] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015. [Online]. Available: https://doi.org/10.1038/nature14236

[42] P. Zhu, X. Li, and P. Poupart, "On improving deep reinforcement learning for pomdps," *CoRR*, vol. abs/1704.07978, 2017. [Online]. Available: http://arxiv.org/abs/1704.07978

[43] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[44] U. Kaytaz, S. Ucar, B. Akgun, and S. Coleri, "Distributed deep reinforcement learning with wideband sensing for dynamic spectrum access," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, May 2020. [Online]. Available: https://doi.org/10.1109/wcnc45663.2020.9120840

[45] Y. Xu, J. Yu, W. Headley, and R. Buehrer, "Deep reinforcement learning for dynamic spectrum access in wireless networks," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*. IEEE, Oct. 2018. [Online]. Available: https://doi.org/10.1109/milcom.2018.8599723

[46] F. Li, B. Shen, J. Guo, K.-Y. Lam, G. Wei, and L. Wang, "Dynamic spectrum access for internet-of-things based on federated deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7952–7956, Jul. 2022. [Online]. Available: https://doi.org/10.1109/tvt.2022.3166535

[47] J.-M. Kang, "Reinforcement learning based adaptive resource allocation for wireless powered communication systems," *IEEE Communications Letters*, vol. 24, no. 8, pp. 1752–1756, 2020.

[48] A. Kaur, J. Thakur, M. Thakur, K. Kumar, A. Prakash, and R. Tripathi, "Deep recurrent reinforcement learning-based distributed dynamic spectrum access in multichannel wireless networks with imperfect feedback," *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 2, pp. 281–292, Apr. 2023. [Online]. Available: https://doi.org/10.1109/tccn.2023.3234276

[49] K. Yang, C. Shen, and T. Liu, "Deep reinforcement learning based wireless network optimization: A comparative study," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2020, pp. 1248–1253.

[50] M. Alrabeiah and A. Alkhateeb, "Deep learning for tdd and fdd massive mimo: Mapping channels in space and frequency," 2019. [Online]. Available: https://arxiv.org/abs/1905.03761

[51] Y. Hua, R. Li, Z. Zhao, X. Chen, and H. Zhang, "Gan-powered deep distributional reinforcement learning for resource management in network slicing," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 334–349, 2020.

[52] Z. Li, C. Guo, and Y. Xuan, "A multi-agent deep reinforcement learning based spectrum allocation framework for d2d communications," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE Press, 2019, p. 1–6. [Online]. Available: https://doi.org/10.1109/GLOBECOM38437.2019.9013763

[53] W. Ning, X. Huang, K. Yang, F. Wu, and S. Leng, "Reinforcement learning enabled cooperative spectrum sensing in cognitive radio networks," *Journal of Communications and Networks*, vol. 22, no. 1, pp. 12–22, 2020.

[54] H. Chen, H. Zhao, L. Zhou, J. Zhang, Y. Liu, X. Pan, X. Liu, and J. Wei, "A dueling deep recurrent math xmlns="http://www.w3.org/1998/math/MathML" id="m1" miq/mi /math-network framework for dynamic multichannel access in heterogeneous wireless networks," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–14, Oct. 2022. [Online]. Available: https://doi.org/10.1155/2022/9446418

[55] Z. Fu, W. Xu, Z. Feng, X. Lin, and J. Lin, "Throughput analysis of LTE-licensed-assisted access networks with imperfect spectrum sensing," in *2017 IEEE Wireless*

Communications and Networking Conference (WCNC). IEEE, Mar. 2017. [Online]. Available: https://doi.org/10.1109/wcnc.2017.7925651