Artificial Intelligence for Spectrum-Aware Autonomous Wireless Networks

Nada AbdelKhalek

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

March 2025

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Miss Nada AbdelKhalek**

Titled: **Artificial Intelligence for Spectrum-Aware Autonomous Wireless Networks**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality. Signed by the Final Examining Committee:

_____ Chair
Dr. Jamal Bentahar

_____ External Examiner
Dr. Abdallah Shami

_____ External to the Program
Dr. Amr Youssef (CIISE)

_____ Examiner
Dr. Mohammad Reza Soleymani

_____ Examiner
Dr. Dongyu Qiu

_____ Thesis Supervisor
Dr. Walaa Hamouda

Approved by _____
Dr. Jun Cai, Graduate Program Director
Department of Electrical and Computer Engineering

24-04-2025 _____
Dr. Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Artificial Intelligence for Spectrum-Aware Autonomous Wireless Networks

Nada AbdelKhalek, Ph.D.

Concordia University, 2025

Spectrum is one of the most vital public resources, carrying wireless communications for mobile phones, satellites, and emergency services. Traditionally, spectrum has been exclusively licensed, resulting in occasional underutilization. However, with the rapid proliferation of wireless devices, spectrum congestion has become inevitable, especially in unlicensed networks such as the Internet of Things (IoT), vehicular, and Unmanned Aerial Vehicles (UAVs), where devices rely on a limited number of public frequency bands that may not support large-scale communications. This contradiction between licensed spectrum underutilization and unlicensed spectrum congestion necessitates a rethinking of spectrum allocation and management strategies. In this thesis, we draw inspiration from Cognitive Radio (CR) technology, which equips radio devices with capabilities such as perception, reasoning, and judgment, and extend it to "intelligent radio" that integrates both cognition and learning capabilities. Our work advocates a shift from traditional model-driven approaches that rely on domain knowledge and strong assumptions to data-driven methods that learn directly from raw data and constant interactions with the environment. With the support of Artificial Intelligence (AI), we design intelligent spectrum borrowing and spectrum-sharing techniques that enable wireless devices to operate opportunistically on licensed bands. Specifically, this thesis explores how AI can endow wireless devices with *context-awareness*, *self-optimization*, and *self-management* capabilities for tasks such as dynamic spectrum access, power management, resource allocation, and ensuring security. Additionally, we develop solutions and frameworks for *self-sustaining* wireless devices that leverage Energy Harvesting (EH), bringing us closer to the

realization of green networks. Our AI-driven algorithms are designed with computational efficiency in mind to minimize the burden on resource-constrained devices.

To drive context-aware intelligence in large-scale cooperative networks, we develop various unsupervised Machine Learning (ML) approaches for spectrum sensing. Unlike existing methods, the proposed frameworks operate without the need for labeled data, prior knowledge of the radio environment, or cooperation between licensed and unlicensed users. The approach ensures robust spectrum sensing while minimizing computational overhead for unlicensed users with limited capabilities. Moreover, we investigate how dimensionality reduction can improve computational efficiency and model generalizability. We expand the use of unsupervised learning to hybrid CR networks to allow devices to detect all licensed network states, opening up new opportunities for dynamic spectrum access.

To improve spectrum reasoning and analysis, we introduce some of the first fully unsupervised, data-efficient deep representation learning frameworks. These frameworks are designed to learn effective and disentangled representations of radio environment data. We demonstrate their effectiveness in significantly enhancing spectrum gap detection in small-scale cooperative networks. Additionally, we tackle key challenges of unsupervised learning, such as sensitivity to initialization and the need for predefined cluster counts. In large-scale networks, we propose a generative deep representation model that not only learns efficient representations but also captures the distribution of radio environment data, enabling the generation of new, unseen samples.

To facilitate edge intelligence and enhance the privacy of intelligent radios, we propose the first fully unsupervised deep Federated Learning (FL) framework for secure and distributed spectrum sensing in large-scale mobile networks. By leveraging user mobility across a large geographical area, the method enhances spatio-temporal diversity without requiring the transmission of private data to a central unit for processing. Instead, data is collected locally, and a shared model is collaboratively trained in a decentralized manner, significantly reducing communication overhead and safeguarding user privacy.

We tackle the growing challenge of spectrum scarcity in Cognitive IoT (CIoT) networks, where the demand for spectrum is increasing due to the expansion of connected devices. To address this, we develop intelligent and adaptive control algorithms for the joint management of network

resources in spectrum-sharing environments. First, we formulate optimization problems under various constraints and model the decision-making process of a CIoT agent in the dynamic radio environment. We then propose two novel Deep Reinforcement Learning (DRL) algorithms that enable devices to autonomously learn operational strategies to optimize network resources and maximize long-term throughput without comprehensive prior knowledge. Additionally, we introduce innovative exploration strategies to enhance the CIoT agent's ability to identify optimal actions that maximize data rates. Considering the resource limitations of these networks, the algorithms are designed to be lightweight to reduce computational burdens on users. We also integrate EH techniques, such as Wireless Power Transfer (WPT) and Simultaneous Wireless Information and Power Transfer (SWIPT), to make these networks self-sustaining.

Finally, to develop dynamic strategies for navigating hostile spectrum-sharing environments impacted by jamming attacks, we propose an intelligent DRL approach that does not rely on frequency hopping. This algorithm is designed for rapid convergence, energy efficiency, and adaptability to adversarial conditions. We begin by formulating the optimization problem of power control under various constraints and modeling the decision-making process of the CIoT agent in such a hostile environment. Then, we introduce a novel interference-aware exploration strategy that enables the CIoT device to autonomously learn a transmission strategy, effectively mitigating jamming attacks and maximizing performance. Furthermore, we leverage WPT EH to allow the CIoT agent to convert jamming interference into a valuable resource for recharging.

In summary, the contributions of this thesis lay the foundation for a new generation of intelligent, autonomous wireless networks that are both spectrum-aware and agile, capable of optimizing resources and adapting to dynamic and complex environments.

Acknowledgments

Today, I stand on the shoulders of giants whose knowledge and guidance have laid the path that has brought me to this moment.

I am especially grateful to my supervisor, Prof. Walaa Hamouda, for his guidance and mentorship, which have been the backbone of my academic journey. Prof. Hamouda, you have been a beacon of support, always there to offer advice when needed, but also providing the space for me to develop my own ideas. Our meetings and conversations meant so much to me, for they sparked new ideas, challenged my thinking, and helped shape my work. Your thoughtful critiques and insightful questions refined my research, while your encouragement pushed me beyond what I thought was possible. During difficult moments when I doubted myself, you always took the time to listen, provide constructive feedback, and remind me of my potential. Your dedication to my success has played a defining role in shaping the young researcher I am today, and I will always be grateful for your trust in me and for inspiring me to reach my highest potential.

I also would like to thank my PhD committee members, Prof. Dongyu Qiu, Prof. M. Reza Soleymani, and Prof. Amr Youssef, for their invaluable contributions to my PhD journey. Your insightful feedback consistently challenged me to elevate the quality of my work, and I am deeply thankful for your support throughout these years. A special thanks to Prof. M. Reza Soleymani, who has been an exceptional professor to me during both my master's and PhD. He has taught me so much, and I will forever be grateful for his guidance and kindness. I would also like to extend my thanks to Dr. Andrew Delong, who was formerly a member of my PhD advisory committee. His passion for excellence in teaching and research has been a great source of inspiration for me to propel forward in the field of machine learning. I would like to sincerely thank Prof. Karim Seddik

for encouraging me to pursue my PhD dream early in my undergraduate years.

I am deeply grateful to the *Natural Sciences and Engineering Research Council (NSERC)* and the *Gina Cody School of Engineering and Computer Science* at Concordia for their generous support in partially funding my research. My heartfelt thanks also go to the *Fonds de Recherche Québec – Nature et Technologies (FRQNT)* for awarding me their doctoral fellowship early in my PhD journey. This support went beyond financial assistance; it was a meaningful recognition by the province of Québec of the value of my research, motivating me to stay focused and excel. Merci beaucoup pour votre soutien, pour avoir cru en la valeur de mes recherches et pour m'avoir offert un environnement où je peux m'épanouir. I also extend my sincere appreciation to the *Gina Cody School of Engineering and Computer Science*, the *School of Graduate Studies*, the *Graduate Students' Association (GSA)*, the *IEEE Women in Communications Engineering (WICE)*, and the *IEEE Communications Society (ComSoc)* for enabling me to present my work at top-tier international conferences. Their support has been instrumental in shaping my academic and professional growth.

A heartfelt thank you to Audrey Veilleux, our Graduate Program Coordinator, for being an incredible source of support. Your constant willingness to help, always with a smile and always just an email away, has been a true blessing. I deeply appreciated your open-door policy and the way you made every challenge feel manageable. It was also a pleasure working with you to organize the *Graduate Students Research Conference (GSRC)*—it was an experience I thoroughly enjoyed, and your leadership made it a true success!

I would like to extend my heartfelt thanks to all my students at Concordia throughout the years, whose enthusiasm, curiosity, and passion have made teaching such a rewarding experience. You have not only inspired me, but also helped shape me into a better educator and a better learner. I will always cherish the thoughtful and fun discussions we had, and I deeply value the insights and perspectives you brought to every class.

I am deeply grateful to my friends, who have become my second family in Montréal. Through every high and low, you have been there, offering your love, encouragement, and support. The memories we have created together will stay with me forever. I would like to thank, in no particular order, the wonderful women whose friendship has been a constant source of strength and joy, Hadeer Elashhab, Nadia Abdolkhani, Salma Elmahallawy, and Anastasiia Kulyk. I am so fortunate to have

each of you in my life, and I will always treasure the bond we share. I would also like to thank my friends from back home, Dina Gamal, Salma Emara, Omar Khouly, and Abdallah Omran, for always cheering me on and encouraging me to stay positive. I would like to sincerely thank Mohamed Gharbia for always lightening the hardest of days, and for always being just an arm's length away whenever I needed help.

Last but not least, I would like to thank my dear mother, Maha Kanz. Without you, I would not be the person I am today. You have been my greatest source of strength, always pushing me to excel, to persevere, and to believe in myself. Your endless love and devotion have shaped me into a better human. Thank you for standing by me, for trusting in my journey, and for inspiring me with your kindness and resilience. I can only hope to one day be the kind of parent you are. This thesis is dedicated to you.

To my guiding light, my beloved mother, Maha Kanz,

whose endless love, faith, and nurturing have made me who I am today.

In the memory of my dear grandparents, Kanz Abdelrehim & Safia Hassan.

To all the women who dare to dream.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

β-VAE β-Variational Autoencoder

Adam Adaptive Moment Estimation

AI Artificial Intelligence

AP Affinity Propagation

AS Antenna Switching

ASR Average Sum Rate

AUC Area Under the ROC Curve

BP Backpropagation

CIoT Cognitive IoT

CNN Convolutional Neural Network

CR Cognitive Radio

CSS Cooperative Spectrum Sensing

D3QN Dueling Deep Q-Network

DDQN Double Deep Q-Network

DL Deep Learning

DNN Deep Neural Network

DQN Deep Q-Network

DRL Deep Reinforcement Learning

DSA Dynamic Spectrum Access

DT Decision Tree

EH Energy Harvesting

EM Expectation-Maximization

FC Fusion Center

FL Federated Learning

GMM Gaussian Mixture Model

IoT Internet of Things

LR Logistic Regression

LSTM Long Short-Term Memory

MARL Multi-Agent Reinforcement Learning

MDP Markov Decision Process

ML Machine Learning

MSE Mean Squared Error

PCA Principal Component Analysis

PDF Probability Density Function

PR Precision-Recall

PS Power Splitting

PU Primary User

QoS Quality of Service

ReLU Rectified Linear Unit

RF Random Forest

RL Reinforcement Learning

RNN Recurrent Neural Network

ROC Receiver Operating Characteristics

Rx Receiver

SAE Sparse Autoencoder

SGD Stochastic Gradient Descent

SNR Signal-to-Noise Ratio

SU Secondary User

SVM Support Vector Machine

SWIPT Simultaneous Wireless Information and Power Transfer

TS Time Switching

Tx Transmitter

Tx-Rx Transmitter-Receiver

UCB Upper Confidence Bound

VAE Variational Autoencoder

WPT Wireless Power Transfer

List of Common Symbols

$\boldsymbol{\mathcal{A}}$	Action space
$\boldsymbol{\mathcal{P}}$	Set of state transition probabilities
$\boldsymbol{\mathcal{R}}$	Set of possible rewards
$\boldsymbol{\mathcal{S}}$	State space
η	Learning rate
γ	Discount factor
$\hat{\lambda}_a^t$	Actual-expected jammer interference
$\hat{r}_a^t$	Expected reward
$\nu_{m,n}$	Multi-path fading component between the m-th PU and n-th SU
ω	Primary channel bandwidth
$\overline{U}_a^t$	Actual-expected reward
$\psi_{m,n}$	Shadow fading component between the m-th PU and n-th SU
ρ_m	Transmit power of the m-th PU
B_t	CIoT battery level at time slot t
B_{max}	Battery capacity of CIoT Tx
c'	UCB algorithm hyperparameter

C_m^{PU} Position coordinate of the m-th PU

C_n^{SU} Position coordinate of the n-th SU

e_t Energy harvested in time slot t

E_{max} Maximum possible energy to be harvested

$g_{m,n}$ Power attenuation between the m-th PU and n-th SU

g_{ps}^t Channel power gain for the PU Tx and CIoT Rx

g_{sp}^t Channel power gain between the CIoT Tx and PU Rx

g_{ss}^t Channel power gain for the CIoT Tx-Rx pair

$h_{m,n}$ The channel gain between the m-PU and the n-th SU

I_{th} Interference threshold

m Number of primary users

n Number of secondary users

N_n Thermal noise at the n-th SU

P_j^t Jammer transmit power in time slot t

P_p^t PU transmit power in time slot t

P_s^t CIoT Tx transmit power in time slot t

R_t Achievable rate of the CIoT Tx in time slot t

X_m Signal transmitted by the m-th PU

Chapter 1

Introduction

1.1 Overview

The evolution from Morse code to Sixth-Generation (6G) networks has profoundly transformed human communication. Over the past three decades, the rapid advancement of wireless technologies has driven numerous innovations that have reshaped daily life. From mobile devices and connected vehicles to drone operations, emerging technologies have been on the rise. These developments pave the way for an intelligently connected future with spectrum-intensive applications. While the evolution of wireless networks continually reshapes modern society, the consequences of these innovations on spectrum congestion are becoming increasingly evident. The demand for frequency spectrum is rising exponentially as the number of connections in next-generation wireless networks, including Fifth-Generation (5G) and beyond, continues to grow. These networks aim to offer seamless access to diverse communication services, such as immersive metaverse experiences, deep-sea exploration, and non-terrestrial communications. For instance, recent forecasts suggest that the number of connected Internet of Things (IoT) devices will increase from 15.9 billion in 2023 to 18.8 billion by 2025 [1]. This rapid expansion is fueling an unprecedented demand for wireless communication resources, emphasizing the critical importance of efficient spectrum management. Furthermore, the report [1] indicates that this surge in connectivity and high data rate demands is expected to persist throughout the next decade, with no signs of abating. Consequently, fixed spectrum allocation strategies are becoming increasingly unfeasible, resulting in significant

inefficiencies in spectral utilization. To overcome these challenges, dynamic spectrum access and management solutions are essential.

It has been nearly 30 years since Joseph Mitola III first introduced Cognitive Radio (CR) technology, which was envisioned to create "brain-powered" communications to address the challenges of spectrum utilization efficiency. By equipping wireless devices with CR technology, they can opportunistically borrow or share licensed spectrum bands. All that is required is an excellent perception-action decision-making process, allowing devices to understand their surroundings and efficiently borrow/share licensed spectral resources without interfering with licensed users. That is, a CR system is one that possesses awareness of its spectral environment and responds to statistical variations with two primary objectives: ensuring reliable communication and optimizing spectral utilization efficiency. According to Ericsson's latest analysis of emerging technology trends in wireless communications, CR is expected to play a pivotal role in next-generation networks by embedding autonomy at the core of network operations [2]. However, mere environment awareness and responsiveness are not sufficient for a CR node to be genuinely *cognitive*. True cognition requires the capability to learn from past experiences and adjust behavior accordingly. This is where Artificial Intelligence (AI) becomes integral to CR networks. By leveraging AI, devices can actively and intelligently observe their surroundings, learn from environment patterns—such as channel dynamics and licensed user activity—and make informed decisions that enhance their ability to efficiently borrow or share spectral resources.

Learning is a crucial component of intelligent radio systems, particularly in scenarios where the exact relationship between inputs and outputs is unknown. In such cases, learning techniques can be utilized to approximate the system's input-output function. For instance, in wireless communications, channels are inherently non-optimal and are highly dynamic. Consequently, learning-based approaches enable nodes to adapt efficiently even without complete awareness of environment characteristics or network topology. Moreover, when nodes operate in unfamiliar environments, conventional decision-making techniques may be impractical due to their reliance on extensive system information. As an example, the CR decision-making cycle can be formulated as a Markov Decision Process (MDP), which is traditionally addressed using dynamic programming. However, optimal solutions can be achieved through learning techniques such as Reinforcement Learning

(RL) [3–6] without requiring knowledge of the model's state transition probabilities. Learning-driven optimization offers *self-management* capabilities for a wide range of challenges related to network optimization and resource management [7]. Additionally, adopting low-complexity learning methods, which are characterized by their simple structure and computational efficiency, can substantially reduce the complexity of intelligent radio systems.

1.2 Motivation

While cognitive systems have been extensively studied over the past three decades, most research has relied on model-driven approaches that are heavily dependent on prior knowledge, strong assumptions, or mathematical modeling. However, these methods come with inherent limitations. For example, if the environment undergoes significant changes, the model may fail, and if users operate in an unfamiliar context, the approach can become ineffective. To address these challenges, this thesis shifts from model-driven to data-driven approaches, leveraging raw radio environment data to enhance adaptability and performance. The central question driving the research presented in this thesis is: *How can we create a wireless communication system that operates autonomously in a dynamic environment by only relying on raw collected data?* We envisioned an intelligent system capable of passively gathering data and, from that data alone, constructing its own operation models and strategies. These would enable the system to optimize network aspects such as dynamic spectrum access, power management, resource allocation, and security. In summary, the key motivations for this thesis are:

(1) To develop data-efficient learning approaches that equip radio devices with AI-powered *context-awareness*, allowing them to dynamically borrow licensed spectrum and identify unused portions effectively by relying solely on raw collected spectrum data.

(2) To explore data-driven approaches that learn efficient radio data representation, without prior knowledge or strong assumptions, aiming to improve spectrum reasoning and analysis capabilities of intelligent radio devices.

(3) To design intelligent and adaptive control algorithms, alongside the use of energy harvesting

technologies, that empower radio devices with *self-optimization*, *self-management*, and *self-sustaining* capabilities, enabling them to perform in unknown or hostile dynamic spectrum-sharing environments, without comprehensive precise knowledge.

Several studies have applied learning techniques to perform cognition tasks. However, most rely on supervised learning, which requires labeled data—i.e., input features X paired with corresponding outcome labels Y. To obtain these labels, some works assume communication with licensed users, which contradicts the core principles of CR, while others rely on prior environment knowledge to manually label the data. Neither of these approaches is practical for CR systems. Even if feasible with increasing data volumes, these methods are not scalable. It has been shown that increasing the number of cooperating, spatially diverse unlicensed users to collect spectrum data can improve spectrum hole detection. However, as the number of users grows, the data dimensionality increases, making it computationally expensive to train learning algorithms on such data. Furthermore, this data often contains redundancies, which reduce efficiency. Therefore, there is a need to develop mechanisms that preprocess spectrum data effectively to enhance the learning model's capacity. Additionally, considerations must be made for unlicensed users with limited capabilities. In some cases, there may be a need to offload decision-making to a central entity with higher computational capacity, which can process the data and learn an effective model to perform CR tasks. Given these challenges, there is an urgent need for unsupervised, data-efficient learning frameworks that enable frequency-domain context-awareness, allowing radio devices to intelligently identify and access available spectrum.

Much of the existing research in CR has focused on designing expert-tuned functions to model, shape, or adapt signals in complex radio environments. However, Deep Learning (DL) approaches offer an alternative by allowing systems to autonomously extract meaningful features and uncover hidden patterns in data, improving spectrum reasoning and analysis. Despite these advantages, many prior works rely on large volumes of labeled data or computationally heavy architectures, placing significant burdens on end users. Additionally, many approaches rely on extensive cooperation between unlicensed users to detect available spectrum gaps, which can lead to significant communication overhead and inefficient use of resources. On the other hand, when unlicensed

networks have fewer users, the available degrees of freedom are significantly reduced, making it essential to explore how representation learning can boost performance in such situations. In other cases, reasoning tasks are offloaded to centralized units or cloud servers, particularly in large-scale networks. While this can enhance performance, it also introduces security vulnerabilities. These challenges highlight the need for lightweight, scalable, and secure data-driven approaches that can efficiently learn to automatically extract valuable features from radio environment samples, thereby enhancing spectrum analysis.

Traditional offline optimization methods for resource management and allocation often struggle in dynamic or large-scale environments, limiting their effectiveness. To enable *self-managing* and *self-optimizing* networks, intelligent algorithms are needed to autonomously improve power management, data rates, and security through direct interaction with the environment. However, special considerations must be made when designing such algorithms for resource-constrained unlicensed users, particularly to avoid overstraining their computational power and energy resources. Cooperative learning-based optimization methods require a common objective among users and rely on centralized training, making them vulnerable to security threats. Additionally, they are unsuitable for ad-hoc scenarios where users frequently join and leave the network. Distributed methods, such as those enabled by Multi-Agent Reinforcement Learning (MARL) frameworks, also encounter convergence issues due to the need for state information exchange, which adds signaling overhead. As the number of unlicensed users increases, these methods become less scalable. Similarly, Reinforcement Learning (RL)-based non-cooperative strategies face scalability challenges as the number of environment states and potential actions grows. Finally, the openness of the radio environment creates vulnerabilities, especially when both attackers and unlicensed users are confined to a single channel by the licensed network. Mitigating these attacks while ensuring smooth network operations, avoiding resource strain, and maximizing gains is a significant challenge. These gaps highlight the need for novel, robust intelligent approaches to resource management and allocation in dynamic radio environments, enabling energy-constrained devices to operate autonomously, sustainably, and securely.

1.3 Contributions

In light of the prevailing contradiction between spectrum scarcity and congestion and the preceding discussions, this thesis aims to explore the potential of Artificial Intelligence (AI) frameworks and Energy Harvesting (EH) mechanisms to equip radio devices with *context-awareness*, *self-optimization*, *self-management*, *self-sustaining* capabilities, enabling them to optimize dynamic spectrum access, power management, resource allocation, and security. The contributions of this thesis are summarized as

ML-Driven Context-Awareness for Enhanced Spectrum Sensing. In Chapter 3, we focus on developing unsupervised learning approaches tailored for spectrum sensing in large-scale cooperative networks. Unlike existing learning-based methods, our proposed frameworks operate without the need for labeled data, prior knowledge of the radio environment, or cooperation between licensed and unlicensed users. We begin by analyzing a system model for large-scale cooperative networks and subsequently propose several unsupervised learning frameworks with two key objectives: (1) ensuring robust spectrum sensing and (2) minimizing computational overhead for unlicensed users with limited capabilities. In this context, we investigate techniques for training supervised models using unsupervised data, leveraging their superior performance without the need for labeled samples. To further enhance computational efficiency and model generalizability, we explore dimensionality reduction techniques for unsupervised learning. Finally, we extend the applicability of unsupervised learning to hybrid CR networks. Unlike prior studies, which primarily focus on detecting idle or busy states, we demonstrate how unsupervised learning can enable devices to identify the full range of licensed network states, unlocking new possibilities for dynamic spectrum access. Our findings demonstrate that our proposed unsupervised ML approaches achieve performance comparable to supervised learning benchmarks without relying on labeled data, prior knowledge, or cooperation with the licensed network.

Deep Representation Learning for Advanced Spectrum Reasoning and Analysis. In Chapter 4, we explore unsupervised deep representation learning frameworks to equip intelligent radio devices with advanced spectrum reasoning and analysis capabilities—without relying on prior knowledge or large volumes of training data. We first examine their application in small-scale cooperative

networks, where only a few users collaborate for spectrum gap detection, thereby limiting the network's degrees of freedom. However, we demonstrate that our proposed unsupervised Deep Learning (DL) approaches, which learn effective representations of sensing data, can significantly boost detection performance in such constrained settings. Additionally, we address key challenges associated with unsupervised learning, including sensitivity to cluster centroid initialization and the need for predefined cluster counts. Next, we extend our study to large-scale cooperative networks, where our proposed method not only learns disentangled representations but also develops a generative model capable of producing new, unseen samples by capturing the underlying distribution of sensing data in a latent space. Extensive simulations across diverse network configurations, propagation environments, and fading conditions validate the effectiveness of our approach. Additionally, our proposed methods achieve performance comparable to supervised DL-based techniques while surpassing non-DL methods, underscoring their potential for robust spectrum analysis.

Distributed Learning in Large-Scale Mobile Spectrum-Aware Networks. In Chapter 5, we design the first fully unsupervised deep Federated Learning (FL) framework for robust, distributed, and secure spectrum sensing in large-scale cooperative mobile networks. Our approach leverages user mobility across a vast geographical area to enhance spatio-temporal diversity. Unlike prior works, our method does not require users to transmit private data to a central unit or cloud. Instead, users collect data locally and collaboratively train a shared model in a fully decentralized manner. This decentralized approach significantly reduces the communication overhead associated with transmitting sensing data to a central unit for spectrum gap identification. At the same time, it enhances the model's generalization capacity and safeguards user privacy, giving individuals control over their own data. Our proposed framework is data-efficient and does not require large amounts of training samples. Moreover, our results demonstrate the effectiveness and scalability of the proposed approach, highlighting its superior performance compared to existing DL and FL methods for spectrum sensing.

Intelligent Algorithms for Adaptive Control and Resource Allocation in Spectrum Sharing Networks. In Chapter 6, we tackle the spectrum scarcity challenge in Cognitive IoT (CIoT) networks, which face an ever-growing demand for spectrum due to the exponential increase in connected devices each year. In this context, we focus on two key challenges: (1) joint power control and channel

access coordination in Wireless Power Transfer (WPT)-enabled CIoT networks and (2) joint time and power management in Simultaneous Wireless Information and Power Transfer (SWIPT)-enabled CIoT networks. Unlike prior works, our approach considers the competing interests of users in the network, their limited computational capacity, and their energy constraints. To enhance practicality, we incorporate realistic energy harvesting mechanisms that do not rely on a dedicated, stable source. We formulate the optimization problems under multiple constraints, including channel occupancy, competition, channel gain, energy arrival, battery capacity, and interference. To address these problems, we first model the decision-making process of the CIoT agent as a model-free Markov Decision Process (MDP). We then propose two Deep Reinforcement Learning (DRL) algorithms that enable the agent to navigate the dynamic spectrum-sharing environment without requiring prior knowledge or assumptions. These algorithms allow the agent to learn an effective strategy for solving the joint optimization problems while maximizing the long-term achievable rate. To ensure efficiency, our DRL algorithms are designed to be lightweight, minimizing computational overhead for the user. Additionally, we incorporate novel exploration strategies to enhance the agent's ability to discover optimal actions that maximize data rates. We benchmark our proposed approaches against existing DRL methods in the literature, demonstrating their ability to converge to a stable state across various simulation settings while significantly outperforming baseline approaches.

Dynamic Strategies for Navigating Hostile Spectrum Sharing Environments. In Chapter 7, we explore hostile spectrum-sharing environments impacted by jamming attacks. Most existing works focus on frequency hopping or power control strategies to mitigate such threats. However, these approaches may not always be viable. In scenarios where unlicensed users are confined to a single shared channel, frequency hopping is not an option. Additionally, under spectrum-sharing constraints, unlicensed users must carefully regulate their transmission power to remain below the interference threshold tolerated by licensed users. Furthermore, power control strategies for energy-constrained devices must be carefully designed to prevent excessive energy consumption. To address these challenges, we first formulate the throughput optimization problem while considering key factors such as jamming attacks, channel gain, and interference constraints. We then propose an intelligent DRL algorithm that enables a CIoT agent to autonomously navigate hostile spectrum-sharing environments and learn an optimal transmission strategy to maximize its own performance.

Our algorithm is designed for fast convergence, improving energy efficiency and ensuring rapid adaptability to adversarial conditions. Additionally, we consider a WPT-enabled network, allowing the CIoT agent to harvest energy from jamming signals–transforming interference into a beneficial resource rather than a harmful obstacle. To further enhance decision-making, we introduce a novel exploration strategy that efficiently balances action discovery, maximizing user gains while minimizing jammer interference. Our results demonstrate that, even in the presence of jamming attacks, the proposed algorithm can dynamically switch between data transmission and energy harvesting while performing power control to optimize network operations. Furthermore, our DRL approach achieves its objectives with considerable success, significantly outperforming benchmarks.

The contributions mentioned above have been published in [8–16] or accepted for publication in [17]. Additional research conducted during my PhD tenure has been published in [18].

1.4 Thesis Organization

The remainder of this thesis is structured as follows. Chapter 2 provides the necessary background on key concepts that form the foundation of this work. Chapter 3 explores unsupervised Machine Learning (ML) frameworks that enable network users to develop frequency-domain context awareness, allowing for more robust and reliable spectrum sensing. In Chapter 4, we investigate deep representation learning techniques through unsupervised DL for advanced spectrum reasoning and improve the analytical capabilities of intelligent radio devices. Chapter 5 extends this discussion to distributed learning, where we design an unsupervised deep Federated Learning (FL) framework to strengthen spectrum awareness in large-scale mobile networks. In Chapter 6, we focus on intelligent and adaptive control in energy-harvesting CIoT networks, leveraging DRL to tackle two critical challenges: (1) optimizing joint power control and channel access coordination, and (2) managing joint time and power allocation for efficient network operation. Building on this, Chapter 7 examines how DRL can enable CIoT networks to make autonomous, intelligent decisions that enhance security in hostile spectrum-sharing environments while simultaneously improving performance and extending network lifetime. Finally, Chapter 8 concludes the thesis with a summary of our findings, key insights, and potential directions for future research.

Chapter 2

Background

2.1 Brain-Powered Communications

Modern and future wireless networks are confronted with the pressing challenge of spectrum scarcity and congestion. The continuous influx of data streams from numerous real-time monitoring devices has already led to, and is expected to further drive, an exponential surge in wireless traffic. Over time, the fixed spectrum allocation policy enforced by regulatory authorities, such as the Federal Communications Commission (FCC), has raised significant concerns regarding spectrum under-utilization. The paradox of licensed spectrum scarcity and unlicensed spectrum congestion has spurred the advancement of opportunistic spectrum access, also referred to as Dynamic Spectrum Access (DSA), enabled by Cognitive Radio (CR) technology. Under DSA, unlicensed Secondary Users (SUs) opportunistically share or reuse spectrum bands originally assigned to licensed Primary Users (PUs). Notable examples of licensed spectrum include TV white spaces and cellular frequency bands [19].

The three primary CR access mechanisms are overlay, underlay, and interweave. Each CR access type has unique characteristics and design considerations. In an overlay CR network, SUs must collaborate with PUs to gain spectrum access rights. For instance, SUs can relay PUs' transmissions, allowing them to utilize the licensed band for their own data transmission. In underlay CR networks, simultaneous spectrum access is permitted as long as SUs' transmissions do not interfere with PUs' communication. Consequently, SUs must regulate their transmit power to ensure they remain within

the interference threshold of the PU receiver [20]. In an interweave CR network, SUs are only allowed to access PUs' bands when they are unoccupied, meaning concurrent transmissions with PUs are not permitted. Under this scheme, SUs first perform spectrum sensing to acquire radio environment information and identify available spectrum gaps [21]. If PUs reappear, SUs must vacate the channel and search for alternative idle bands [22]. Fig. 2.1 illustrates the interweave and underlay CR access models and their interactions in both time and frequency domains.

Before the emergence of the CR networking paradigm, perception and reasoning mechanisms were not typically incorporated into wireless network architectures. Instead, terminal firmware relied on hard-coded rules to guide observations and actions. The CR cycle consists of three key phases: spectrum sensing, spectrum analysis, and spectrum decision. The first phase, *perception*, is achieved through spectrum sensing and monitoring, enabling the CR node to detect ongoing activities in its environment. Spectrum sensing is a crucial step, as the accuracy and reliability of the sensed information directly impact subsequent stages in the cycle. Furthermore, SUs must monitor the spectrum bands to promptly detect the sudden reappearance of PUs. Next, the CR node utilizes its *reasoning* capability to interpret the sensed data during the spectrum analysis phase, where an SU

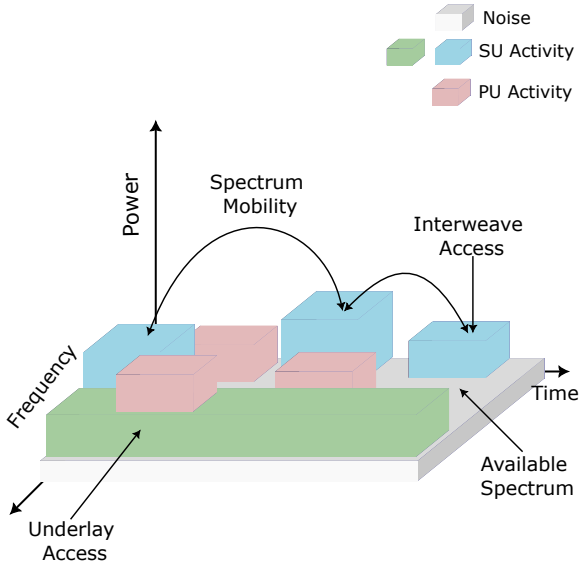


Figure 2.1: Illustration of underlay and interweave access models in cognitive radio. In the interweave model, SUs transmit only when PUs are inactive, as shown by the gaps in frequency and time. In contrast, the underlay model allows SUs to transmit concurrently with SUs, subject to power constraints, as demonstrated by overlapping regions of frequency and time.

evaluates and processes the collected information. The final phase, spectrum decision, involves the CR node employing its *judgment* abilities to determine which band the SU should utilize based on its properties and user requirements. In addition to selecting the communication band, the SU also configures transmission parameters, including modulation type and data rate [23]. As noted in [24], intelligence is defined by three core conditions: perception, learning, reasoning, and judgment. Consequently, for a CR to derive reasoning and judgment from perception, it must possess learning capabilities. Learning requires that current actions be informed by both past and present observations of the environment, making prior experiences a critical factor in the CR cognition cycle.

2.2 From Cognitive to Intelligent Radio

The term *cognitive* encompasses concepts such as awareness, perception, thinking, and decision-making. As previously mentioned, for a CR node to derive reasoning and judgment from perception, it must have the capacity to learn. Learning involves making decisions based on both past and present observations of the environment. In the context of CR, Machine Learning (ML) enables devices to actively recognize patterns and behaviors across various network topologies and modalities, leading to improved network performance in new and unfamiliar environments without prior knowledge [25,26]. This adaptability is a fundamental trait of intelligence. Traditional optimization techniques in CR are not naturally suited to adapt to new and unforeseen circumstances, as they often depend on fixed models and parameters, which are ill-equipped for dynamic environments or unexpected events. Furthermore, traditional methods typically rely on mathematical models that require approximations due to system complexity. In contrast, ML empowers the CR network to learn directly from historical data, eliminating the need for approximations [27]. This allows the system to optimize multiple network parameters simultaneously in an adaptive manner, a task that is challenging for traditional optimization approaches [28].

Intelligent radio extends beyond merely learning to derive reasoning and judgment; it incorporates several characteristics that allow it to be truly intelligent. Below are some of the key features that define intelligent radio.

Firstly, by harnessing their intelligence and predictive capabilities, intelligent radios effectively

process large volumes of sensing and monitoring data, resulting in the development of a detailed knowledge map of the spectral environment. This knowledge serves as a crucial asset for optimizing multiple network parameters simultaneously, thus enhancing the network's ability to access and utilize available spectral resources. Moreover, intelligent radios can forecast future events, such as shifts in traffic patterns [29], network congestion [30], energy harvesting potential [31], and spectrum availability [32], based on the acquired knowledge.

Secondly, intelligent CR includes an advanced resource management framework that tackles various challenges in different wireless networks, such as power control and resource allocation. In contrast to traditional distributed optimization methods, which are typically performed offline or in a semi-offline fashion, ML enables intelligent radios to coordinate decentralized actions in real-time, fostering more efficient collaboration [33]. This dynamic decision-making capability, bolstered by continuous learning and improvement, is a core feature of intelligent radio. At times, CRs must conduct a search or optimization process to determine the best configuration for a given environment. Although these tasks can be time-consuming and computationally expensive, ML models provide a solution by learning and storing past case-solution pairs, facilitating faster decision-making in the future [34]. Thus, the intelligence provided by ML enables adaptive and responsive behavior, which is critical for next-generation wireless communications.

Thirdly, ML-enabled devices can exhibit *self-organizing*, *self-healing*, and *self-optimizing* abilities, which help address a variety of challenges, such as security and energy efficiency. Intelligent radios can autonomously enhance their security and privacy by effectively identifying and mitigating different types of smart attacks, thus improving the resilience and reliability of wireless networks. For example, CRs have the ability to answer a critical question: *When should it be recognized that the existing learning model is no longer suitable for the new environment and needs updating?* The solution lies in detecting current signals while leveraging previously acquired knowledge. CR nodes can learn the typical behavior of the radio environment, and when deviations from this learned behavior occur, the CR network can deduce the possible presence of security threats [35]. Additionally, ML-driven CR can optimize energy consumption and storage by intelligently harvesting energy from various sources based on their availability, such as PUs, the radio environment, or ambient sources. For instance, energy-harvesting CRs systems can develop the ability to fine-tune

their transmission strategies to maximize data transfer rates. This allows them to select their energy source, choosing between SUs and PUs depending on their availability [36]. This adaptive energy management approach significantly extends the battery life of wireless devices in networks like IoT, reducing the need for frequent battery replacements or recharging, which can be costly and impractical in challenging environments. Thus, ML-based CRs can operate more efficiently and sustainably, ensuring optimal performance and resource utilization.

Fourthly, the integration of ML in CR is particularly beneficial in large and heterogeneous network environments, overcoming the constraints of traditional centralized and distributed optimization methods that struggle with network scale and diversity. By utilizing learning techniques such as Reinforcement Learning (RL), CR agents can factor in multiple crucial elements, such as dynamic and unpredictable channel conditions, when making decisions. This comprehensive approach, as opposed to focusing on individual components or engaging in complex joint optimization, enables CR agents to pursue both their individual and collective objectives [37]. As a result, ML-driven CR facilitates broader and more efficient utilization of available spectrum resources, leading to enhanced network performance across varied environments.

Finally, an important aspect of intelligent radio systems is the ability to learn and understand the behavior and preferences of PUs in spectrum-sharing scenarios, such as overlay or hybrid modes. By analyzing and modeling the interactions between PUs and SUs, intelligent radios can adjust their spectrum access strategies to maximize the efficient use of available spectrum resources. By learning the behavior [38] and preferences of PUs [39], intelligent radios can make more informed decisions regarding spectrum allocation, power control, and access protocols. This enables a tailored and efficient use of the spectrum, considering the specific requirements and usage patterns of the primary network. For example, in overlay CR, SUs can develop the ability to balance their own transmissions with the provision of relaying services to PUs. This decision-making process can consider several factors, such as the available spectrum, channel quality, and expected network lifetime [39, 40]. As a result, both the primary and secondary networks can benefit from enhanced performance, reduced interference, and improved overall spectral efficiency.

2.3 Artificial Intelligence Frameworks

Learning frameworks can be categorized into four types depending on the nature of the training data or environment: supervised learning, unsupervised learning, and reinforcement learning.

2.3.1 Supervised Learning

Given a set of inputs and their corresponding outputs, the result of applying a supervised learning algorithm can be represented as a function $y = f(\mathbf{x})$, which takes a new input $\mathbf{x}$ and generates an output y. The exact form of the function $y = f(\mathbf{x})$ is determined during the training phase, also known as the learning phase. Supervised learning problems are applicable when CR nodes have some prior knowledge about the environment, often captured in a labeled data format. This learning type is also referred to as learning by *instruction* [24], where the training data consists of input feature vectors $\mathbf{x}$ and their corresponding target labels y, enabling the learning algorithm to estimate the function $y = f(\mathbf{x})$. In interweave CR networks, supervised learning is occasionally applied to determine whether a channel is vacant or not [41]. These types of problems are known as *classification* problems. Examples of supervised learning algorithms include Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF), among others.

2.3.2 Unsupervised Learning

In unsupervised learning, the training data consists of input vectors $\mathbf{x}$ without any corresponding target labels y (i.e., unlabeled data). The goal of this type of learning is often to find clusters of similar samples within the data, a process referred to as *clustering*, or to estimate the distribution of the input space, known as *probability density estimation*. Unsupervised learning is particularly suitable for CR nodes operating in unknown environments. In such cases, autonomous unsupervised learning enables CR nodes to explore the environment properties and *self-adapt* without requiring prior knowledge [11]. Algorithms such as k-means and Gaussian Mixture Model (GMM) are common examples of unsupervised learning methods. Additionally, unsupervised learning can be employed for dimensionality reduction, transforming high-dimensional input spaces into lower-dimensional

ones for *visualization* purposes. Techniques like t-Distributed Stochastic Neighbor Embedding (T-SNE) and Principal Component Analysis (PCA) are examples of such algorithms. Fig. 2.2 illustrates the key differences between supervised and unsupervised learning models.

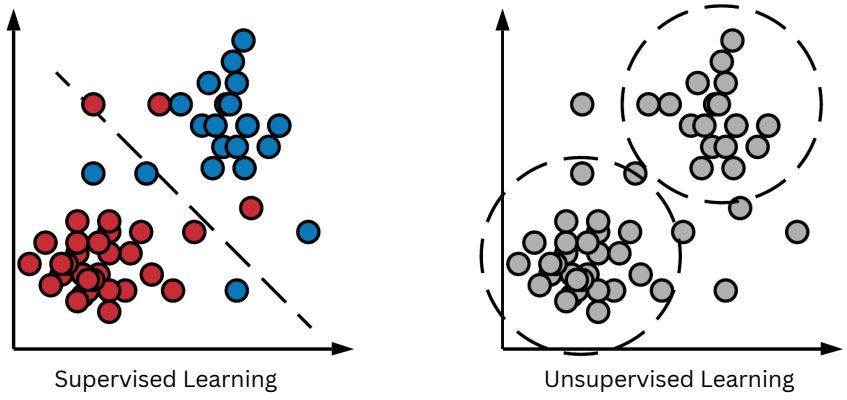


Figure 2.2: Comparison of supervised and unsupervised learning models: On the left, supervised learning is illustrated with data points colored according to their labels, where the model learns a classification boundary. On the right, unsupervised learning is shown, where all points are uncolored due to the absence of label information, requiring the model to group the data based on inherent patterns.

2.3.3 Reinforcement Learning

Reinforcement Learning (RL) focuses on determining the appropriate actions an agent should take within a given environment to maximize its reward. Unlike supervised learning, where the algorithm is provided with explicit optimal outputs, RL requires the agent to discover the optimal actions through *trial-and-error*, also referred to as model-free learning. As shown in Fig. 2.3, the agent interacts with its environment by transitioning between states and taking actions. The reward, which is feedback from the environment, can be either positive or negative, signaling whether the agent's actions were beneficial or not. In many cases, the immediate action affects not only the immediate reward but also future rewards across all subsequent time steps. A key concept in RL is the trade-off between *exploration* and *exploitation*. Exploration involves the agent trying new behaviors to assess their effectiveness, while exploitation focuses on using known actions that maximize the reward. Excessive emphasis on either exploration or exploitation can lead to

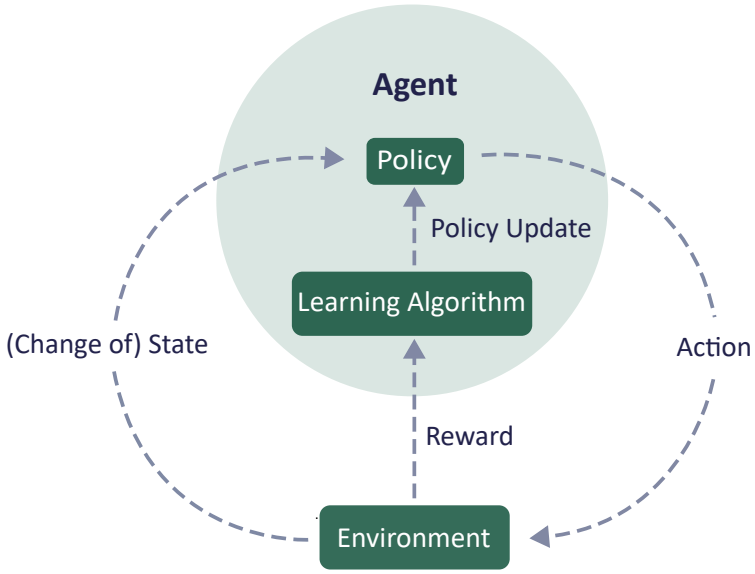


Figure 2.3: Illustration of the key components of the reinforcement learning process, including the agent, environment, states, actions, rewards, and the learning mechanism. The agent interacts with the environment by taking actions based on the current state, receiving feedback in the form of rewards or penalties, and updating its policy to maximize cumulative reward over time.

suboptimal results. RL is widely used for decision-making within CR networks. Some popular Reinforcement Learning (RL) algorithms include, but are not limited to, Q-learning, Actor-Critic (AC), and Deep Deterministic Policy Gradient (DDPG).

2.3.4 Deep Learning

DL is a branch of ML that draws inspiration from our understanding of the human brain, statistics, and applied mathematics. At its core, DL enables devices to learn and recognize patterns through a hierarchical process of concept-building. Central to the DL approach are neural networks, also referred to as Multilayer Perceptron (MLP) or Artificial Neural Network (ANN). A neural network, illustrated in Fig. 2.4, consists of multiple interconnected layers of *neurons*, and when a neural network contains more than one layer, it is called a Deep Neural Network (DNN). DNNs can be employed to carry out tasks such as classification or regression, for example, predicting the actions of primary users in a CR network to optimize goals such as Quality of Service (QoS), reliability, and throughput. Various neural network architectures exist, including Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), each suited for specific data types. DL can

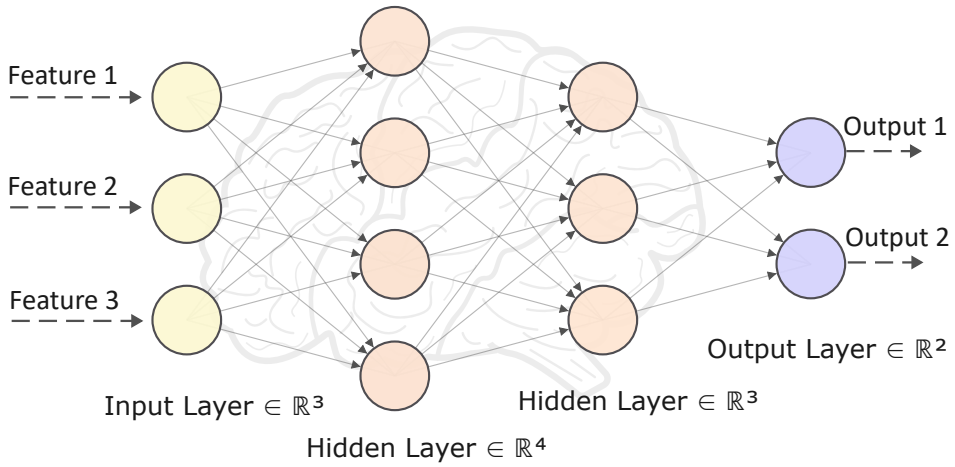


Figure 2.4: Illustration of the architecture of a feed-forward fully connected deep neural network. The structure of the network consists of an input layer, two hidden layers, and an output layer. Each hidden layer consists of multiple neurons that apply a non-linear activation function to the weighted sum of inputs.

be applied in both unsupervised and supervised settings. In unsupervised learning, approaches such as representation learning or dimensionality reduction are used when unlabeled data is available. Common unsupervised DL architectures include Autoencoders (AEs), Variational Autoencoders (VAEs), and Generative Adversarial Networks (GANs). On the other hand, when labeled data is accessible, several basic DL network architectures, such as CNN, RNN, and Long Short-Term Memory (LSTM), can be employed.

2.3.5 Federated Learning

Federated Learning (FL) has introduced transformative changes as a distributed machine learning approach. Recently developed by Google [42], FL enables coordination among various devices to perform ML training without the need to share raw data, ensuring privacy protection and reducing network resources required for data transmission. The concept of federated learning is depicted in Fig. 2.5. The local model parameters are sent to the edge/cloud server for global model aggregation, and the updated global model parameters are then transmitted back to the devices. As a result, FL consumes fewer communication resources than centralized learning since only changes in model

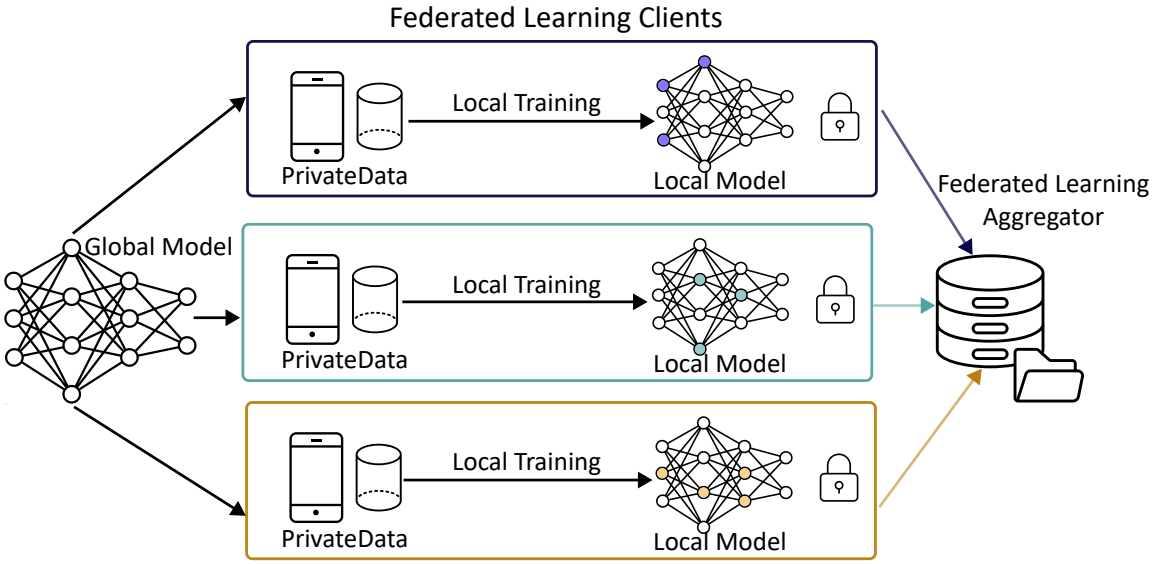


Figure 2.5: Illustration of the federated learning framework, where multiple decentralized devices (clients) collaboratively train a shared learning model while keeping their data local. Each client performs local model updates based on its data and sends only the updated model parameters to a central server. The server aggregates the updates from all clients and refines the global model, which is then sent back to the clients for further training.

parameters are exchanged rather than entire datasets. Various emerging wireless networks have adopted FL to address critical issues such as security and reliability [43, 44], energy consumption, and communication costs [45], as well as utilization efficiency and fairness [46], paving the way for its integration into intelligent radio networks.

FL can greatly enhance learning scalability and address security vulnerabilities by enabling many devices to train a learning model in parallel. This is especially valuable in situations where the dataset is too large to be stored or processed centrally, or when data privacy concerns prevent sharing. By allowing radio devices to train the model locally and only transmit small updates to a central unit, FL reduces communication overhead and facilitates model training with extensive spectrum data [43, 44]. Additionally, FL optimizes the learning process by enabling the model to learn from diverse data sources. This approach improves the model's generalizability and reduces the likelihood of overfitting to a particular data distribution. Moreover, the model can continuously improve as new data is gathered from various devices, eliminating the need for a central authority to collect, store, and process all the data. These benefits make FL a promising solution for enhancing the performance and security of wireless networks, underscoring the importance of addressing privacy

and security concerns when designing intelligent radio systems.

2.4 Security and Privacy Threats to Intelligent Radio

In intelligent radio systems, security and privacy are critical due to the inherent vulnerabilities of wireless communication channels. Major threats include jamming, eavesdropping, and the falsification of sensing data, all of which can interfere with network operations, jeopardize the confidentiality of transmitted data, and compromise the accuracy of sensing information. Below, we explore some of the key attacks that present significant challenges in maintaining secure, reliable, and trustworthy communication within intelligent radio networks.

2.4.1 Jamming Attacks

Jamming attacks are a significant threat to the physical layer of radio networks. In these attacks, a jammer typically emits a high-powered signal over one or more frequency bands for varying durations. These signals disrupt critical communications and reduce the Signal-to-Noise Ratio (SNR) at the receivers. To mitigate such attacks, the use of frequency-hopping spread spectrum techniques, which involve the transmitter rapidly and randomly switching between multiple channels, has been strongly advised. Additionally, various jamming detection methods, including fuzzy logic, game theory, and channel surfing, have been proposed in the literature.

2.4.2 Eavesdropping

Users within the broadcast range of a transmitter may be able to intercept shared secret communications. These users are referred to as eavesdroppers and are typically classified into active and passive categories. Active eavesdroppers are identifiable users within the network who are unauthorized and untrustworthy, allowing them to obtain channel state information (e.g., TV transmission). On the other hand, passive eavesdroppers, which are more common and realistic to consider, attempt to intercept private communications without emitting harmful signals. While passive eavesdroppers are often discussed in the literature, obtaining accurate wiretapping channel state information remains challenging, particularly when the location of the passive eavesdropper is unknown.

2.4.3 Spectrum Sensing Data Falsification Attacks

A spectrum sensing data falsification attack is a security threat in CR networks where an attacker transmits false or deceptive spectrum sensing data to a CR node, aiming to disrupt or manipulate network operations. For instance, an attacker could provide inaccurate information to a Fusion Center (FC), leading it to make erroneous decisions regarding the availability of frequency channels. This could disrupt legitimate communication or even grant the attacker unauthorized access to the network. Such attacks are particularly harmful in critical systems, like emergency services, where reliable communication is essential for public safety.

2.5 Energy Harvesting for Self-Sustaining Networks

Reducing energy consumption and carbon emissions is vital for the advancement of green communications, particularly in wireless networks that rely on energy-constrained devices. The frequent need for battery replacements is not only costly and impractical but also environmentally damaging, making sustainable energy solutions necessary for long-term network sustainability. Energy Harvesting (EH) plays a crucial role in green communications, enabling CR users to extend battery life and maintain network continuity. The process of EH involves harvesting energy from various renewable sources, such as solar, wind, vibration, and radio frequency waves [47]. This process converts Alternating Current (AC) signals into Direct Current (DC) signals (electricity) to power devices. EH offers numerous advantages, making it an attractive solution for various CR networks. For example, it allows IoT devices and sensors to become *self-sufficient*, eliminating the need for external recharging or battery replacements. Additionally, EH is particularly beneficial in scenarios where using batteries is challenging or impractical, such as powering sensors and monitoring devices in remote areas or emergency situations, where traditional power lines or battery changes are unfeasible. It can also be applied to small electronics in wearables, smart homes, and IoT devices. Furthermore, EH reduces dependency on non-renewable energy sources and lowers carbon emissions [48], contributing to a reduction in fossil fuel reliance and advancing towards a more sustainable future.

2.5.1 Energy Harvesting Technologies

EH in wireless communication networks can be achieved through various methods, with WPT and SWIPT being among the most widely used. WPT allows network nodes to recharge their batteries by harnessing electromagnetic radiation. In WPT, green energy can be harvested in two ways: from ambient signals or a dedicated power source, such as a base station, that provides controlled energy. SWIPT is an advanced version of WPT that facilitates the simultaneous transmission of both energy and data. However, the efficient implementation of SWIPT necessitates significant modifications in the design of wireless communication networks. Traditionally, network performance is assessed based on reception reliability and data transfer rates. With SWIPT, a key challenge is managing the trade-off between the information rate and the energy harvested, as users extract energy from radio signals. In a SWIPT system, performing EH and information decoding concurrently on the same received signal is generally impractical, as the EH process tends to interfere with the signal's data content.

To implement SWIPT in practice, the received signal must be split into two separate parts. That is, the receiver should be designed with an energy harvester circuit designed to carry out either Time Switching (TS), Power Splitting (PS), Antenna Switching (AS). Below, we explain these protocols:

- **Power Splitting (PS) Protocol:** The received signal's power is divided into two parts based on a predefined power splitting factor. A portion of the power is allocated for energy harvesting and stored in a battery or capacitor. The remaining power is used for information decoding.

- **Time Switching (TS) Protocol**: The entire received power is utilized within each time slot, but the slot is divided between energy harvesting and information decoding. The duration allocated to each function is determined by a time switching factor. Both WPT and SWIPT play a vital role in enhancing the sustainability of wireless networks by providing efficient energy solutions for autonomous and battery-constrained devices.

- **Antenna Switching (AS) Protocol:** A subset of the available antennas is allocated for EH, while the remaining antennas are used for information decoding. In contrast to TS and PS, AS is simpler and more attractive for practical SWIPT architecture designs.

Chapter 3

Context-Aware Intelligence for Enhanced Spectrum Sensing

3.1 Introduction

Cognitive devices use their perception and reasoning abilities to monitor the radio environment and determine if the licensed channel is idle or busy in interweave access mode, as they can only use the spectrum when licensed users are not actively using it. As a result, the main challenge in such access mode is the accurate detection of PU activity through spectrum sensing. Traditional spectrum sensing techniques include feature detection, matched filter detection, and energy detection [18]. However, these are model-driven methods that require prior knowledge of the environment and often rely on strong assumptions that may not always be valid in real-world situations. In contrast, underlay CR access permits SUs to share the spectrum with the primary network, as long as they maintain interference within a tolerable threshold set by the PUs. The hybrid underlay-interweave CR approach combines the strengths of both techniques. In such systems, SUs must adjust their transmission parameters dynamically when PUs are active to minimize interference. When the spectrum is idle, they can transmit at maximum power until the PUs reappear [49]. This interference threshold is not fixed; it varies according to the primary network's activity. By accurately detecting this activity, SUs can optimize their spectrum usage–maximizing transmission power when feasible and enhancing overall network performance.

To tackle the above, this chapter moves from traditional model-driven approaches to data-driven methods, enabling intelligent radio devices to learn from their environment. Specifically, we explore unsupervised Machine Learning (ML) frameworks that enable CR networks to develop frequency-domain context awareness. This capability empowers users to perform robust Cooperative Spectrum Sensing (CSS) in both interweave and hybrid CR networks.

3.2 Related Works

For cognitive users to operate effectively in dynamic spectrum environments, they must move beyond traditional rule-based methods and incorporate learning-driven decision-making. Artificial Intelligence (AI) stands out as a powerful enabler, providing radio devices with the capability to autonomously analyze, adapt, and optimize spectrum usage. Various AI techniques have been explored in the literature on CR networks, including but not limited to Machine Learning (ML), DL, and RL [18]. A comparative study in [50] examined different learning methods for performing CSS and demonstrated that the Support Vector Machine (SVM) algorithm outperforms all benchmark algorithms. Research like [51] explored the performance of learning-based CR networks that use energy levels as features for spectrum sensing. In [52], a Gaussian Mixture Model (GMM) was utilized for spectrum sensing in mobile environments. A spectrum sensing approach in [53] employed a neural network that primarily relied on unlabeled data, with a small amount of labeled data gathered when the PU were absent. In [54], a recurrent neural network was proposed for spectrum sensing, and its performance was compared to that of SVM. Studies such as [41, 55, 56] have focused on detecting PU activity in hybrid CR networks using supervised ML and DL.

Firstly, although CR systems have been extensively researched, several challenges remain, especially in the context of learning-based CR networks. A significant issue is that many existing studies rely on supervised learning for spectrum sensing, which requires labeled data–i.e., spectrum data that is paired with its channel occupancy state. Consequently, the effectiveness of supervised learning methods is heavily reliant on the availability of labeled training data. For example, [57] assumes that the primary network occasionally sends labeled data to the secondary network, a scenario that not only contradicts the core principles of CR, but also introduces considerable SU-PU

communication overhead. Similarly, [53] assumes access to a small amount of labeled data gathered when the PUs are inactive, which requires prior knowledge of their inactivity. Moreover, supervised learning approaches depend on large amounts of labeled data, further intensifying the SU-PU communication overhead. While these methods have shown strong performance, they come with notable practical limitations. As a result, we believe there is a simpler and more feasible way to achieve the performance of supervised learning-based spectrum sensing without the need for labeled data.

Secondly, to the best of our knowledge, there has been limited research on the development of unsupervised learning-based sensing strategies for hybrid CR networks. Most existing studies, such as [55, 56], focus on detecting PU activity in hybrid CR networks. However, these studies primarily concentrate on distinguishing between busy and idle states within the primary network, which simplifies the spectrum sensing task. In hybrid CR networks, users must identify which PUs are active at any given time in order to adjust their transmission parameters and avoid interference. Additionally, works such as [41, 56] use supervised ML and DL techniques for spectrum sensing, which require labeled data for training. This dependence on labeled data presents practical challenges, highlighting the need for unsupervised learning approaches for more effective sensing.

3.3 Contributions

Motivated by the aforementioned gaps, we focus on developing unsupervised learning approaches designed for Cooperative Spectrum Sensing (CSS) that do not require any labeled data, prior knowledge about the radio environment, or secondary-primary user communication/cooperation. Our contributions can be summarized as:

- We propose a system model for large-scale cooperative CR networks that addresses two key challenges: (1) ensuring robust spectrum sensing and (2) minimizing computational overhead for SUs, that have limited processing capabilities.
- We introduce an unsupervised learning framework that capitalizes on the prominent performance of supervised models without requiring any labeled data for spectrum hole detection. Specifically, we employ an unsupervised Gaussian Mixture Model (GMM) to learn the latent

structure within the collected spectrum energy data and generate effective labels. These inferred labels are then used to train a supervised Support Vector Machine (SVM) model for accurate channel state prediction.

- To improve the efficiency of unsupervised learning in large-scale networks, where high-dimensional energy data poses a computational challenge, we propose the GMM-PCA framework for CSS. By leveraging Principal Component Analysis (PCA) for dimensionality reduction, this approach significantly enhances both the computational efficiency and training performance of the GMM.

- Finally, we expand the applicability of the GMM-PCA framework to hybrid CR networks, where SUs must distinguish between multiple primary network states beyond idle/busy classification. By leveraging raw radio environment data, we demonstrate the potential of GMM-PCA for multi-class classification in an entirely unsupervised manner.

- We comprehensively evaluate the performance of our proposed unsupervised ML frameworks for CSS and benchmark them against other learning algorithms. Extensive simulations validate their effectiveness in spectrum sensing, demonstrating their relevance across various network settings. To ensure a rigorous assessment, we employ key performance metrics, including Receiver Operating Characteristics (ROC), Area Under the ROC Curve (AUC), Precision-Recall (PR), Area Under the PR Curve (AUPR), training and testing accuracies, and training time. Our results indicate that the proposed unsupervised frameworks for CSS achieve performance comparable to supervised learning benchmarks–all while operating without labeled data or direct SU-PU cooperation.

3.4 System Model

Consider a cooperative CR network consisting of N spatially distributed SUs ($n = 1, ..., N$) and a primary network consisting of M potential PUs ($m = 1, ..., M$). The PUs operate within a dedicated bandwidth ω using a multiple access technique. We adopt a generalized model where the PUs alternate between active and inactive states independently from one another. The SUs and

PUs are scattered over a large geographical area, with the position coordinates of the n-th SU and the m-th PU denoted by C_n^{SU} and C_m^{PU}, respectively. Each SU detects the spectrum's energy level using an energy detector and reports the result to the Fusion Center (FC), which then decides on the spectrum's availability. During a sensing period τ, each SU collects $\omega\tau$ energy samples of the spectrum. The i-th energy sample measured by the n-th SU is the sum of the signals from the active PUs and the thermal noise, represented by

$$E_n(i) = \sum_{m=1}^{M} s_m h_{m,n} X_m(i) + N_n(i), \tag{3.1}$$

where s_m denotes the state indicator for the m-th PU. If the m-th PU is occupying the channel, then s_m is 1; otherwise, it is 0. The channel gain between the m-th PU and the n-th SU is represented by $h_{m,n}$, and $X_m(i)$ is the symbol transmitted by the m-th PU. The thermal noise at the n-th SU, denoted as $N_n(i)$, is modeled as a Gaussian distribution with a mean $\mu_n = 0$ and variance $\sigma_n^2 = \mathbb{E}[|N_n(i)|^2]$. Therefore, the PU detection problem can be framed as a binary hypothesis test, where $H_0 = N_n(i)$ indicates an empty channel, and $H_1 = E_n(i)$ indicates otherwise. Each SU estimates the energy level normalized by the Power Spectral Density (PSD) of the noise as

$$y_n = \frac{2}{\sigma_n^2} \sum_{i=1}^{w\tau} |E_n(i)|^2. \tag{3.2}$$

y_n follows a non-central chi-squared distribution of $q = 2\omega\tau$ degrees of freedom and a non-centrality parameter ζ_n as

$$\zeta_n = \frac{2\tau}{\sigma_n^2} \sum_{m=1}^{M} s_m g_{m,n} \rho_m. \tag{3.3}$$

$g_{m,n} = |h_{m,n}|^2$ is the power attenuation from the m-th PU to the n-th SU given by

$$g_{m,n} = PL(||C_m^{PU} - C_n^{SU}||).\psi_{m,n}.\nu_{m,n}. \tag{3.4}$$

$\psi_{m,n}$ and $\nu_{m,n}$ are the shadow fading and the multi-path fading components respectively, which are assumed to be quasi-static during the time duration of interest.

The path loss component $PL(d) = d^{-\alpha}$ is evaluated based on the Euclidean distance $||.||$ and the path loss exponent α. The transmit power of the m-th PU ρ_m is given by

$$\rho_m = \frac{\sum_{i=1}^{w\tau} E[|X_m(i)|^2]}{\tau}. \tag{3.5}$$

The channel occupancy state of the PUs is represented by a vector $\mathbf{S}$, which contains the states of all M PUs. Given the current channel occupancy state vector $\mathbf{s} = (s_1, ..., s_M)$, if the number of channel samples, i.e., $\omega\tau$, is sufficiently large, the distribution of the energy level y_n at the n-th SU can be modeled as a Gaussian distribution. This distribution has a mean $\mu_{y_n|\mathbf{S}=\mathbf{s}}$ and variance $\sigma^2_{y_n|\mathbf{S}=\mathbf{s}}$ as

$$\mu_{y_n|\mathbf{S}=\mathbf{s}} = E[y_n|\mathbf{S} = \mathbf{s}] = 2w\tau + \frac{2\tau}{\sigma_n^2} \sum_{m=1}^{M} s_m g_{m,n} \rho_m, \tag{3.6}$$

$$\sigma^2_{y_n|\mathbf{S}=\mathbf{s}} = E[(y_n - \mu_{y_n|\mathbf{S}=\mathbf{s}})^2|\mathbf{S} = \mathbf{s}] = 4w\tau + \frac{8\tau}{\sigma_n^2} \sum_{m=1}^{M} s_m g_{m,n} \rho_m. \tag{3.7}$$

All SUs transmit their soft data, i.e., energy levels, to the FC, where an energy vector $\mathbf{y} = (y_1, ..., y_N)$ is constructed. Since each energy level reported follows a normal distribution, the distribution of $\mathbf{y}$ conditioned on the current channel occupancy state $\mathbf{s}$ is modeled as a multivariate Gaussian distribution with the following parameters

$$\boldsymbol{\mu}_{\mathbf{y}|\mathbf{S}=\mathbf{s}} = (\mu_{y_1|\mathbf{S}=\mathbf{s}}, ..., \mu_{y_N|\mathbf{S}=\mathbf{s}}), \tag{3.8}$$

$$\boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{S}=\mathbf{s}} = diag(\sigma^2_{y_1|\mathbf{S}=\mathbf{s}}, ..., \sigma^2_{y_N|\mathbf{S}=\mathbf{s}}). \tag{3.9}$$

$diag(v)$ produces a square matrix with the elements of the vector v arranged along the principal diagonal.

3.5 Unsupervised Learning with Supervised Models for Spectrum Hole Detection

In interweave CR, there is no cooperation between the primary and secondary networks, and SUs must perform blind channel sensing. That is, for each energy vector $\mathbf{y}$, the SUs must determine the channel availability label $d \in \{H_0, H_1\}$. To train a supervised learning classifier, pairs of data points $(\mathbf{y}, d)$ are required. In contrast, an unsupervised learning algorithm only needs a set of energy vectors $\mathbf{Y}$ for training. This leads to the idea of using low-cost unsupervised learning to generate labeled datasets for CR networks. Our aim is to design an unsupervised learning approach that maintains the high performance of supervised models while operating without labeled data. In this section, we present our proposed unsupervised learning framework for Cooperative Spectrum Sensing (CSS), which is entirely based on historically acquired unlabeled data at the Fusion Center (FC). The proposed framework is illustrated in Fig. 3.1. Using this approach, cooperating SUs send their measured energy levels to the FC across L sensing periods, resulting in an unlabeled dataset $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_L\}$ consisting of the energy vectors. The training process begins by feeding the Gaussian Mixture Model (GMM) a set of energy vectors $\mathbf{Y}$ for clustering and labeling. The labels $D = \{d_1, ..., d_L\}$ and the training energy vectors $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_L\}$ are then used to train a SVM classifier, which learns the classification model.

3.5.1 Unsupervised Clustering of Spectrum Data

The GMM is an unsupervised clustering technique that consists of k-multivariate Gaussian distributions superimposed with different weights as follows

$$f(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} v_k.\phi(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{3.10}$$

where $\boldsymbol{\theta}$ denotes all the variables that form the GMM $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, v_k\}$ for $k = 1, ..., K$ Gaussian densities, where $K = 2^M$. The mixing weights must meet $\sum_{k=1}^{K} v_k = 1$ and $v_k \geq 0$. $\phi(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

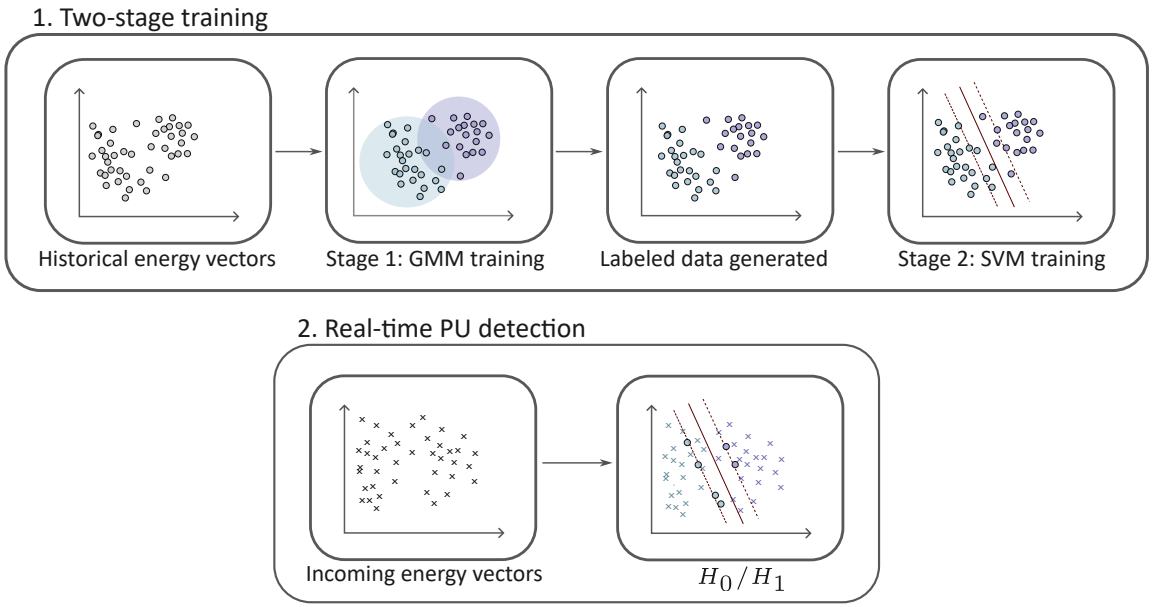


Figure 3.1: The schematic diagram of the proposed unsupervised GMM-SVM framework at the FC.

represents the k-th Gaussian density such that

$$\phi(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_{\boldsymbol{k}}^{-\mathbf{1}}(\mathbf{x}-\boldsymbol{\mu}_k)\}. \tag{3.11}$$

At the FC, a received energy vector $\mathbf{y}$ is drawn from one of two Gaussian distributions: one with a mean vector $\boldsymbol{\mu}_{\mathbf{y}|\mathbf{S}=\mathbf{0}}$ and covariance matrix $\boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{S}=\mathbf{0}}$, or another with mean vector $\boldsymbol{\mu}_{\mathbf{y}|\mathbf{S}=\mathbf{s}}$ and covariance matrix $\boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{S}=\mathbf{s}}$. Given this statistical structure, the GMM is well-suited for modeling the distribution of energy vectors at the FC. In interweave CR networks, the primary focus is on two clusters corresponding to hypotheses H_0 and H_1. The parameters associated with H_0, namely $\boldsymbol{\mu}_{\mathbf{y}|\mathbf{S}=\mathbf{0}}$ and $\boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{S}=\mathbf{0}}$, are known to the CR network before training. However, the parameters $\boldsymbol{\mu}_{\mathbf{y}|\mathbf{S}=\mathbf{s}}$ and $\boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{S}=\mathbf{s}}$, which characterize the k-th Gaussian distribution under hypothesis H_1, remain unknown. Additionally, the mixing proportions v_1 and v_2 are also unknown. Given a set of L training energy vectors, $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_L\}$, the parameters $\boldsymbol{\theta}$ can be estimated using a maximum likelihood approach. The log-likelihood function for $\mathbf{Y}$ is given by

$$\omega(\mathbf{Y}|\boldsymbol{\theta}) = \sum_{l=1}^{L} \ln\left(\sum_{k=1}^{K} v_k.\phi(\mathbf{y}_l|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right). \tag{3.12}$$

A direct computation of (3.12) is infeasible, as it lacks a well-defined optimal maximum. To estimate the set of unknown parameters $\boldsymbol{\theta}$ that maximize the log-likelihood of $\mathbf{Y}$, we utilize the Expectation-Maximization (EM) algorithm–a coordinate descent method applied during training [58]. During the expectation step (E-step), the "responsibility" γ_{lk} is calculated, representing the degree to which each Gaussian component k contributes to an energy vector $\mathbf{y}_l$, as follows

$$\gamma_{lk} = \frac{v_k \phi(\mathbf{y}_l|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j v_j \phi(\mathbf{y}_l|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{3.13}$$

During the maximization step (M-step), the algorithm utilizes γ_{lk} to update the parameter set $\boldsymbol{\theta}$, ensuring that the likelihood of the observed data is maximized. By differentiating the log-likelihood function in (3.12) with respect to $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and v_k, and equating the result to zero, we derive their respective update rules as

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{l=1}^{L} \gamma_{lk} \mathbf{y}_l, \tag{3.14}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{l=1}^{L} \gamma_{lk} (\mathbf{y}_l - \boldsymbol{\mu}_k)(\mathbf{y}_l - \boldsymbol{\mu}_k)^T, \tag{3.15}$$

$$v_k = \frac{N_k}{L}, \tag{3.16}$$

where N_k is the effective number of points in cluster k given as

$$N_k = \sum_{l=1}^{L} \gamma_{lk}. \tag{3.17}$$

After obtaining $\boldsymbol{\theta}$, the GMM computes the log-likelihood of each energy vector as follows

$$\omega(\mathbf{y}_l|\boldsymbol{\theta}) = \ln\left(v_2.\phi(\mathbf{y}_l|\mu_2, \Sigma_2) - \ln v_1.\phi(\mathbf{y}_l|\mu_1, \Sigma_1)\right), \quad \text{for } l = 1, ..., L, \tag{3.18}$$

where $\ln\left(v_2.\phi(\mathbf{y}_l|\mu_2, \Sigma_2)\right)$ is the log-likelihood that energy vector $\mathbf{y}_l$ belongs to cluster H_1. Similarly, $\ln\left(v_1.\phi(\mathbf{y}_l|\mu_1, \Sigma_1)\right)$ is the log-likelihood that energy vector $\mathbf{y}_l$ belongs to cluster H_0. For a decision threshold of δ, if $\omega(\mathbf{y}_l|\boldsymbol{\theta}) \geq \delta$, then $d_l = H_1$, otherwise $d_l = H_0$.

3.5.2 Spectrum Hole Detection Via Support Vector Machines

Following the clustering of spectrum energy vectors using the GMM algorithm, each energy vector $\mathbf{y}_l$ is assigned a corresponding channel occupancy label d_l. This labeled dataset is then utilized to train a Support Vector Machine (SVM) classifier. The primary goal of the SVM is to determine an optimal hyperplane h that effectively separates the training energy vectors. By leveraging support vectors, the SVM maximizes the margin of h while minimizing classification errors. However, due to the inherent noise in the radio spectrum, the collected energy vectors are typically not linearly separable. To overcome this limitation, a non-linear feature-space transformation function $\phi(.)$ is employed to map the energy vectors into a higher-dimensional space. Given a set of L training energy vectors $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_L\}$, the hyperplane h must satisfy

$$\begin{aligned} h(\mathbf{y}_l) = \Psi.\phi(\mathbf{y}_l) + \psi_0 = 0, \\ \text{for } \; l = 1, ..., L. \end{aligned} \tag{3.19}$$

The objective is to determine a weight vector Ψ and a bias term ψ_0, which shifts the hyperplane h away from the origin, ensuring an effective linear separation of the data. However, even after applying the transformation function $\phi(.)$, the hyperplane h may not perfectly separate all training energy vectors due to the inherent complexity of the data distribution. To accommodate misclassified energy vectors and mitigate classification errors, a slack variable ϵ_l is introduced into the model. Consequently, for a given set of L training energy vectors and their corresponding channel state labels $D = \{d_1, \ldots, d_L\}$, the classifier must satisfy

$$\begin{aligned} & d_l[\Psi.\phi(\mathbf{y}_l) + \psi_0] \geq 1 - \epsilon_l, \\ & \epsilon_l \geq 0, \\ & \text{for } \; l = 1, ..., L. \end{aligned} \tag{3.20}$$

ϵ_l is $0 \leq \epsilon_l \leq 1$ for marginal classification errors, and $\epsilon_l > 1$ for misclassification. The convex optimization problem for finding h is constructed as follows [59]

$$\begin{aligned} \text{minimize:} \quad & \frac{1}{2}||\Psi||^2 + \zeta \sum_{l=1}^{L} \epsilon_l \\ \text{subject to:} \quad & d_l[\Psi.\phi(\mathbf{y}_l) + \psi_0] \geq 1 - \epsilon_l, \\ & \epsilon_l \geq 0, \\ & \text{for } \ l = 1, ..., L, \end{aligned} \tag{3.21}$$

where $\zeta > 0$ is a soft margin constant that controls the trade-off between reducing the classification errors and increasing model complexity. Minimizing $\frac{1}{2}||\Psi||^2$ has the same influence as maximizing $2/||\Psi||$, which is the margin of the classifier. Let $\tilde{\beta}_l$ be the solution to the optimization problem in (3.21). The final non-linear decision function is as follows [59]

$$\Gamma(\mathbf{x}) = sgn\left(\sum_{l=1}^{L} \tilde{\beta}_l d_l \kappa(\mathbf{x}, \mathbf{y}_l) + \psi_0 \right), \tag{3.22}$$

where $\kappa(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ represents the kernel function, which computes the inner product of an energy vector and the support vectors in the feature space. An energy vector $\mathbf{y}_l$ qualifies as a support vector if and only if $\tilde{\beta}_l > 0$. The decision function $\Gamma(\mathbf{x})$ can be interpreted as the sum of weighted distances between a test energy vector $\mathbf{x}$ and the support vectors that define h. Common kernel functions include linear, polynomial, and Gaussian kernels [60]. Selecting an appropriate kernel function is crucial, as minimizing the number of support vectors reduces potential classification errors. Once the decision function in (3.22) is obtained, the SVM algorithm classifies a new energy vector $\mathbf{y}^*$ as $d = \Gamma(\mathbf{y}^*)$.

3.6 Dimensionality Reduction for Efficient Spectrum Sensing and Analysis

While the GMM-SVM approach has demonstrated considerable potential in clustering and labeling spectrum sensing data, further improvements can be achieved by refining the clustering

performance of the GMM. As the number of SUs, N, increases, the number of reported energy levels (features) within an energy vector $\mathbf{y}$ also grows. Our previous work [8] highlighted that high-dimensional data can lead to overfitting in the GMM, thereby reducing its generalization capacity. To mitigate this issue, we employ Principal Component Analysis (PCA) to perform dimensionality reduction on the multidimensional energy vectors before training a GMM for CSS. The proposed unsupervised GMM-PCA learning framework at the FC is illustrated in Fig. 3.2.

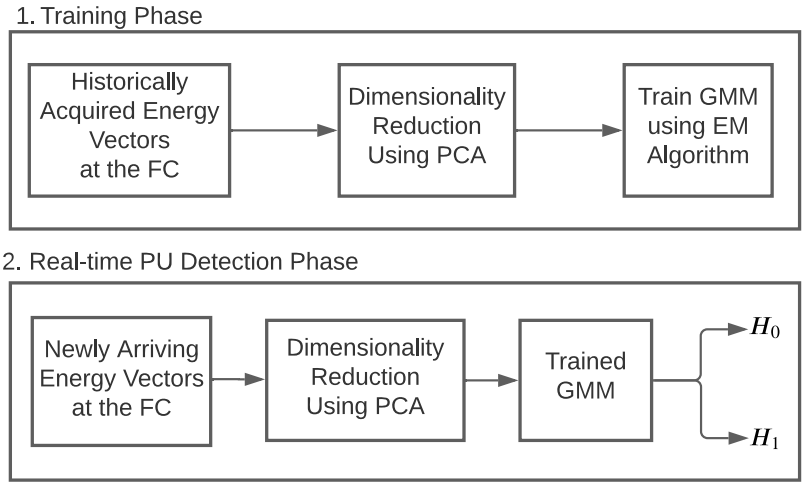


Figure 3.2: Unsupervised sensing at the FC using the GMM-PCA learning approach.

3.6.1 Data Preprocessing Using Principal Component Analysis

PCA identifies the directions of maximum variance in high-dimensional data and projects it onto a K-dimensional subspace using K principal components, thereby preserving most of the radio information [61]. Consider a one-dimensional subspace ($K = 1$) with a direction represented by an N-dimensional vector $\mathbf{u}_1$, where $\mathbf{u}_1^T \mathbf{u}_1 = 1$. Given a set of energy vectors $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_L\} \in \mathbb{R}^{L \times N}$, where N denotes the number of SUs, the sample mean $\overline{\boldsymbol{\mu}}$ and the data covariance matrix $\mathbf{S}$ are defined as

$$\begin{aligned} \overline{\boldsymbol{\mu}} &= \frac{1}{L} \sum_{l=1}^{L} \mathbf{y}_l, \\ \mathbf{S} &= \frac{1}{L} \sum_{l=1}^{L} (\mathbf{y}_l - \overline{\boldsymbol{\mu}})(\mathbf{y}_l - \overline{\boldsymbol{\mu}})^T. \end{aligned} \tag{3.23}$$

The projected data has a mean of $\mathbf{u}_1^T\overline{\boldsymbol{\mu}}$ and variance of $\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1$.

To maximize the variance with respect to $\mathbf{u}_1$, we use a Lagrange multiplier λ_1 then implement an unconstrained maximization such that

$$f(\mathbf{u}_1, \lambda_1) = \mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 + \lambda_1(1 - \mathbf{u}_1^T\mathbf{u}_1). \tag{3.24}$$

By setting the derivative $\frac{\partial f}{\partial \mathbf{u}_1} = 0$ and multiplying by $\mathbf{u}_1^T$, we obtain $\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 = \lambda_1$. This indicates that the variance of the projected data is maximized when $\mathbf{u}_1$ corresponds to the eigenvector associated with the largest eigenvalue λ_1. This result can be extended to cases where $K > 1$ through induction. Consider a K-dimensional subspace defined by the K principal eigenvectors $(\mathbf{u}_1, ..., \mathbf{u}_K)$ of $\mathbf{S}$, along with an additional direction vector $\mathbf{u}_{K+1}$, which is orthogonal to the existing eigenvectors. The orthogonality constraint is enforced using Lagrange multipliers $(\eta_1, ..., \eta_K)$. To maximize the variance $\mathbf{u}_{K+1}^T\mathbf{S}\mathbf{u}_{K+1}$, a Lagrange multiplier λ_{K+1} is introduced, leading to the maximization function in (3.24):

$$g = \mathbf{u}_{K+1}^T\mathbf{S}\mathbf{u}_{K+1} + \lambda_{K+1}(1 - \mathbf{u}_{K+1}^T\mathbf{u}_{K+1}) + \sum_{i=1}^{K} \eta_i \mathbf{u}_{K+1}^T\mathbf{u}_i. \tag{3.25}$$

By setting $\frac{\partial g}{\partial \mathbf{u}_{K+1}} = 0$ and multiplying by $\mathbf{u}_j^T$ for $j = 1, ..., K$, we obtain $\mathbf{S}\mathbf{u}_{K+1} = \lambda_{K+1}\mathbf{u}_{K+1}$. This implies that $\mathbf{u}_{K+1}$ must be an eigenvector of $\mathbf{S}$. Consequently, the result holds for a subspace of $K + 1$ dimensions, thereby completing the inductive step. Thus, it follows that the result is valid for any $K \leq N$.

3.6.2 Leveraging Gaussian Mixture Models for Unsupervised Spectrum Sensing

Given a set of low-dimensional training vectors $\mathbf{z} \in \mathbb{R}^K$ transformed by PCA, the GMM employs the EM algorithm [58], as outlined in Algorithm 1, to estimate the mixture parameters $\boldsymbol{\theta}$ and construct the model in (3.10). Let $\hat{\mathbf{z}} \in \mathbb{R}^K$ represent a low-dimensional testing vector preprocessed by PCA. The GMM calculates the log-likelihood of $\hat{\mathbf{z}}$ using the estimated parameters $\boldsymbol{\Theta}$ as

$$\omega(\hat{\mathbf{z}}|\boldsymbol{\theta}) = \ln\left(v_2.\phi(\hat{\mathbf{z}}|\boldsymbol{\mu_2}, \boldsymbol{\Sigma_2})\right) - \ln\left(v_1.\phi(\hat{\mathbf{z}}|\boldsymbol{\mu_1}, \boldsymbol{\Sigma_1})\right), \tag{3.26}$$

where $\ln\left(v_2.\phi(\hat{\mathbf{z}}|\boldsymbol{\mu_2}, \boldsymbol{\Sigma_2})\right)$ is the log-likelihood that $\hat{\mathbf{z}}$ belongs to cluster H_1. Likewise, $\ln\left(v_1.\phi(\hat{\mathbf{z}}|\boldsymbol{\mu_1}, \boldsymbol{\Sigma_1})\right)$ is the log-likelihood that the $\hat{\mathbf{z}}$ belongs to cluster H_0. For a decision threshold of δ, if $\omega(\hat{\mathbf{z}}|\boldsymbol{\theta}) \geq \delta$, then $d = H_1$, otherwise $d = H_0$.

Algorithm 1 Expectation-Maximization algorithm.

1: **Input:** The set of low dimensional energy vectors $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_L\}$, initial parameters $\boldsymbol{\theta}^{(0)} = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, v_k\}$, number of clusters K
2: **Output:** Estimated parameters $\boldsymbol{\theta}$
3: **Initialize:** Set initial parameter values $\boldsymbol{\theta}^{(0)}$
4: **repeat**
5: **E-Step:** Compute the responsibilities using (3.13) for $l = 1, ..., L$ and $k = 1, ..., K$.
6: **M-Step:** Update the parameters $\boldsymbol{\theta}^{(t)}$ using (3.14), (3.15), and (3.16) for $k = 1, ..., K$.
7: Update iteration counter: $t \leftarrow t + 1$
8: **until** $\boldsymbol{\theta}^{(t)}$ converges
9: **Return:** Estimated parameters $\boldsymbol{\theta}$

3.7 Unsupervised Learning for Situational Awareness in Hybrid Cognitive Radio Networks

In hybrid underlay-interweave CR systems, SUs are permitted to transmit data regardless of PU activity. However, each SU dynamically adjusts its transmission power based on the prevailing channel conditions. During idle spectrum periods, an SU can operate at its maximum allowable power level ρ_{max}. In contrast, when active PUs are present, the SU must regulate its transmission parameters to ensure that the interference does not exceed the designated threshold I_{th}. This interference threshold I_{th} is dictated by the activity status of each PU [62]. Specifically, if the m-th PU is active ($s_m = 1$), the permitted interference level is I_m. Conversely, if the PU is inactive ($s_m = 0$), the interference constraint is considered non-binding. As a result, the interference threshold I_{th} for the primary network is expressed as

$$I_{th} = \min\{I_1, ..., I_M\}. \tag{3.27}$$

The primary network can exist in $C = 2^M$ possible states, where each state c is characterized by a vector $\mathbf{s}_c = (s_1, \ldots, s_M)$, which indicates the activity status of all M PUs. Consequently,

determining the channel state d of the primary network can be formulated as a C-hypothesis testing problem, defined as follows

$$d \triangleq \begin{cases} H_{\mathbf{s}_1} \\ \vdots \\ H_{\mathbf{s}_C} \end{cases} \tag{3.28}$$

where $H_{\mathbf{s}_1}$ indicates an empty channel and $\mathbf{s}_1$ is a zero vector. Each hypothesis $H_{\mathbf{s}_c}$ represents a specific licensed channel state characterized by an activity vector $\mathbf{s}_c$.

3.7.1 Extending the GMM-PCA Approach to Hybrid Cognitive Radio

In hybrid CR networks, acquiring labeled data is often unfeasible due to the lack of cooperation between SUs and PUs, along with the absence of prior knowledge regarding the primary network's channel states. Moreover, spectrum sensing in hybrid CR systems extends beyond distinguishing between idle and occupied spectrum; it also requires identifying active PUs during busy periods [41]. This identification is essential, as the interference tolerance of the primary network depends on the activity status of the PUs [62]. Consequently, by detecting active PUs, SUs can dynamically regulate their transmission power to adhere to the fluctuating interference threshold.

To address this, we formulate the channel state detection problem as an unsupervised clustering task. We extend our proposed GMM-PCA approach to infer the primary network's channel states d based solely on the collected energy levels over L sensing periods, represented as $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_L\}$. To enable SUs to accurately identify all 2^M channel states, we first apply Principal Component Analysis (PCA) (as discussed in Section 3.6.1) to reduce the dimensionality of the energy vectors collected at the FC. This step enhances computational efficiency while preserving the most critical information. Subsequently, we train an unsupervised Gaussian Mixture Model (GMM) using the Expectation-Maximization (EM) algorithm (detailed in Section 3.5.1) to cluster the data into 2^M distinct groups, as opposed to just two groups in interweave CR. To ensure effective clustering, we initialize the cluster mean vectors $\boldsymbol{\mu}_k$ for $k = 1, \ldots, K$ using the K-means algorithm. This method provides a robust starting point by strategically assigning initial cluster centers, which improves both the convergence speed and accuracy of the GMM.

3.7.2 K-means Initialization for Robust GMM Clustering

K-means iteratively partitions the dataset into K clusters by assigning each data point to the nearest centroid $\boldsymbol{\mu}_k$ and updates the centroids until convergence. This assignment is performed using the following rule

$$r_{lk} = \begin{cases} 1 & \text{if} \quad k = \arg\min_j ||\mathbf{y}_l - \boldsymbol{\mu}_j||^2 \\ 0 & \text{otherwise.} \end{cases} \tag{3.29}$$

Accordingly, the cluster centroid $\boldsymbol{\mu}_k$ of each Gaussian density in the GMM is computed as

$$\boldsymbol{\mu}_k = \frac{\sum_{l=1}^{L} r_{lk}\mathbf{y}_l}{\sum_{l=1}^{L} r_{lk}}. \tag{3.30}$$

Additionally, the mixture weights v_k are set proportional to the fraction of data points assigned to each cluster. The K-means clustering algorithm is detailed in Algorithm 2.

Algorithm 2 K-means clustering algorithm.

1: **Input:** The set of low dimensional energy vectors $\mathbf{Z} = \{\mathbf{z}_1, \ldots \mathbf{z}_L\}$, number of clusters K

2: **Output:** cluster centroids $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_K\}$

3: **Initialize:** cluster centroids $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_K\}$ randomly from $\mathbf{Z}$

4: **repeat**

5: **Assignment Step:** Assign each data point to the nearest centroid using (3.29) for $l = 1, \ldots, L$.

6: **Update Step:** Compute new centroids as the mean of assigned points using (3.30) for $k = 1, \ldots, K$.

7: **until** centroids converge

3.8 Simulation Results

In this section, we investigate the performance of a CR network that utilizes our proposed unsupervised ML approaches for CSS.

3.8.1 Setup

A secondary cooperative network is deployed in a geographical area spanning 25 Km^2 in which the SUs are equally spaced. Using a path loss model, the sensing ability is examined, taking into account the impact of large-scale fading. Both the shadow fading $\psi_{m,n}$ and the multi-path fading components $\nu_{m,n}$ are unity. The PUs use the channel with equal probability p and transmit their data independently of each other. The CR network simulation parameters are shown in Table. 3.1.

Table 3.1: Network simulation parameters

CR Parameters		
Parameter	**Symbol**	**Value**
Simulation Area	-	5×5 Km^2
Number of PUs	m	[1,2]
Number of SUs	n	[9:36]
PU Transmit Power	ρ_m	200 mW
Bandwidth	ω	5 MHz
Sensing Period	τ	100 μs
path loss Exponent	α	4
Noise PSD	η	-174 dBm

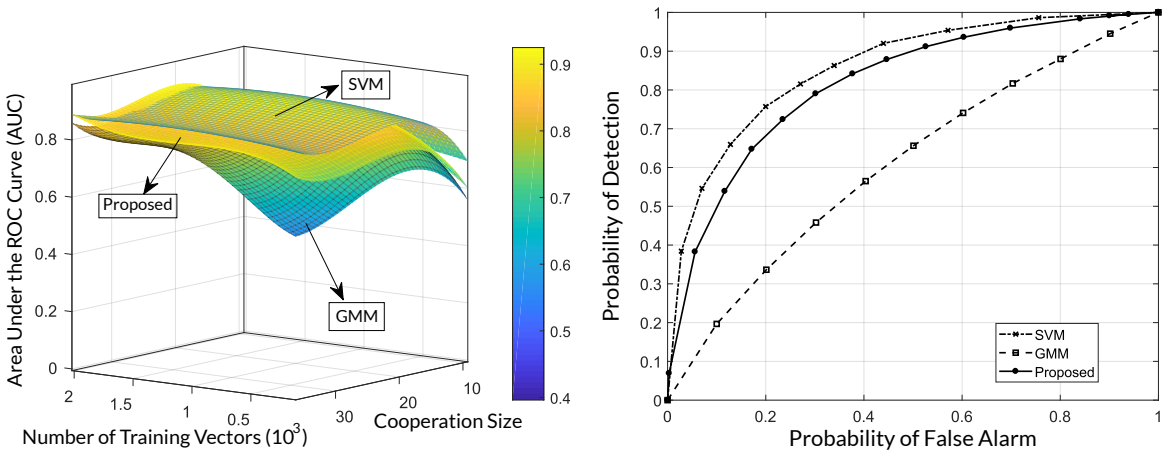
3.8.2 Results and Analysis

Unsupervised Learning with Supervised Models for CSS in Interweave CR

The proposed unsupervised GMM-SVM learning framework relies on the combined performance of the GMM and SVM. The GMM is a computationally inexpensive learning algorithm that does not require labeled data during training; however, it faces challenges with high-dimensional training data. In contrast, the SVM is a supervised learning algorithm that outperforms the GMM but necessitates labeled data. As a result, the detection performance of the proposed learning-based CR network is influenced by both the number of training energy vectors l and the cooperation size n, which corresponds to the dimensionality of the data. To analyze these effects, l is varied from 100 to

2000, and n from 9 to 36, examining their joint impact on detection performance. The performance is evaluated using Receiver Operating Characteristics (ROC) and Area Under the ROC Curve (AUC) as metrics. To fine-tune the hyperparameters of the learning algorithms, a validation set of 800 energy vectors is employed. A linear kernel is chosen for the SVM, as it minimizes the number of support vectors necessary to construct the hyperplane h. Finally, the proposed learning algorithm is tested using 1000 energy vectors to assess performance.

In Fig. 3.3a, it is evident that the SVM outperforms the GMM as it benefits from ground truth, i.e., labeled data. However, the surface of the proposed learning approach is positioned between those of the SVM and the GMM. This can be attributed to the fact that the training process of the algorithm relies on both the SVM and the GMM. As a result, the detection performance of the studied CR network is influenced by the effectiveness of both models. Considering the AUC surface of the GMM in Fig. 3.3a, for high values of n, such as $n = 36$ SUs, the detection performance improves as l increases. This indicates that when the number of high-dimensional training vectors is small, the GMM constructs a suboptimal clustering model and requires additional data samples to enhance learning. When increasing n from 9 to 36 for small values of l, the AUC of the GMM exhibits a concave trend–initially improving due to the inclusion of more spatially diverse SUs, but then declining when the dimensionality becomes excessively high for small values of l.



(a) The joint effect of the number of training vectors l and the cooperation size n for m= 1 PU and C_1^{PU} = (0.5km,0.5km).

(b) The ROC curves at l= 140 vectors, n= 36 SUs, m= 1 PU, and C_1^{PU} = (0.5km,0.5km).

Figure 3.3: Analysis of training energy vectors l and cooperating SUs n on sensing performance, with comparative evaluation of our proposed GMM-SVM approach against other learning techniques.

Conversely, analyzing the SVM surface reveals that increasing the number of SUs enhances detection performance, as the SVM remains robust to high-dimensional data. A summary of the lower and upper performance bounds in terms of AUC is presented in Table 3.2 for $m = 1$ and Table 3.3 for $m = 2$.

In Fig. 3.3b, the GMM, SVM, and the proposed unsupervised learning approach are evaluated at (l=140, n=36, and m=1), which corresponds to a global minimum on the GMM surface. The results indicate that the proposed learning approach achieves detection performance comparable to that of the SVM. In this scenario, the cooperation size is large (n=36), while the number of training energy vectors is relatively small (l=140), leading to a suboptimal clustering model for the GMM. However, the proposed sensing method effectively improves performance. This is because the SVM does not reach a global minimum at (l=140, n=36) due to its resilience to high-dimensional data. As shown in Table 3.2, the global minimum of the SVM surface occurs at (l=140, n=9), as the CR network requires a higher number of spatially diverse SUs to effectively detect PU activity.

Table 3.2: Global minima for the GMM, SVM, and the proposed GMM-SVM approach when $m = 2$ PUs, C_1^{PU}= (0.5km,0.5km), and C_2^{PU}= (-1.5km,0km).

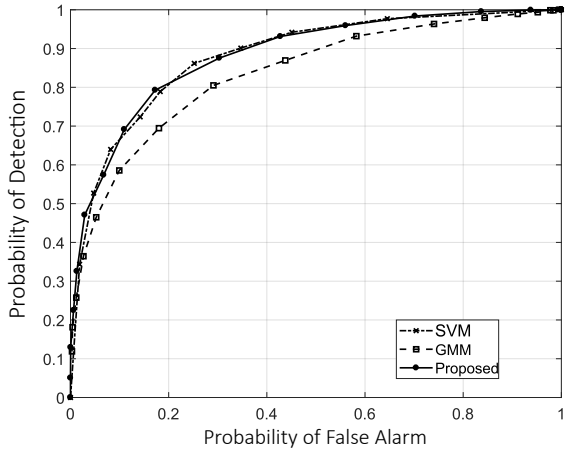
Learning Technique	**Number of Primary Users**	
	m= 1 PU	m= 2 PUs
GMM	(l= 140, n= 36, AUC= 0.5682)	(l= 140, n= 36, AUC= 0.7991)
Proposed	(l= 140, n= 36, AUC= 0.767)	(l= 140, n= 36, AUC= 0.827)
SVM	(l= 140, n= 9, AUC= 0.7043)	(l= 140, n= 9, AUC= 0.888)

Table 3.3: Global maxima for the GMM, SVM, the proposed GMM-SVM learning framework when $m = 2$ PUs, C_1^{PU}= (0.5km,0.5km), and C_2^{PU}= (-1.5km,0km).

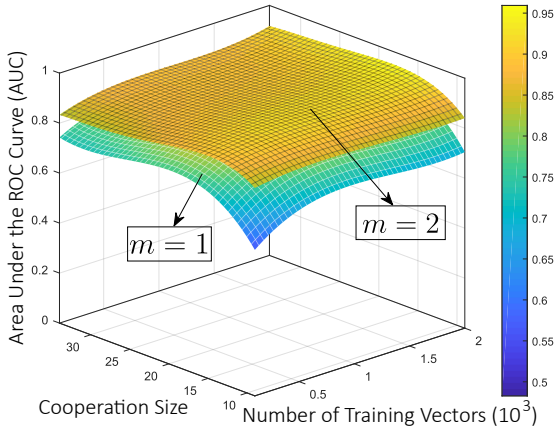
Learning Technique	**Number of Primary Users**	
	m= 1 PU	m= 2 PUs
GMM	(l= 1000, n= 19, AUC= 0.88)	(l= 1964 , n= 23, AUC= 0.897)
Proposed	(l= 1000, n= 19, AUC= 0.901)	(l= 1964 , n= 23, AUC= 0.9459)
SVM	(l= 992, n= 19, AUC= 0.905)	(l= 1960 , n= 23, AUC= 0.9534)

In Fig. 3.4a, the ROC of the GMM, SVM, and the proposed unsupervised GMM-SVM sensing approach are compared at (l=1000, n=19, m=1), which corresponds to a global maximum on the GMM surface. The results show that the proposed method achieves detection performance equivalent to that of the SVM. This is because the GMM constructs a well-fitted clustering model when provided with a sufficient number of training energy vectors (l=1000) and a moderate data dimensionality (n=19). Moreover, the SVM further enhances detection accuracy. By leveraging the strengths of both models, the proposed approach attains the same detection performance as supervised learning.

Examining the detection performance of the CR network under varying numbers of PUs, Fig. 3.4b shows that the surface corresponding to $m = 2$ is elevated compared to the case when $m = 1$. This indicates that as the number of PUs increases, the overall energy levels in the radio environment rise. Consequently, the proposed unsupervised method can more effectively detect the presence of PUs, as the clusters become more distinguishable from one another.



(a) The ROC curves for l= 1000 vectors, n= 19 SUs, m= 1 PU, and C_1^{PU} = (0.5km,0.5km).



(b) The effect of the number of active PUs on detection performance of our proposed unsupervised GMM-SVM approach when m = 2 PUs, C_1^{PU} = (0.5km,0.5km), and C_2^{PU} = (-1.5km,0km).

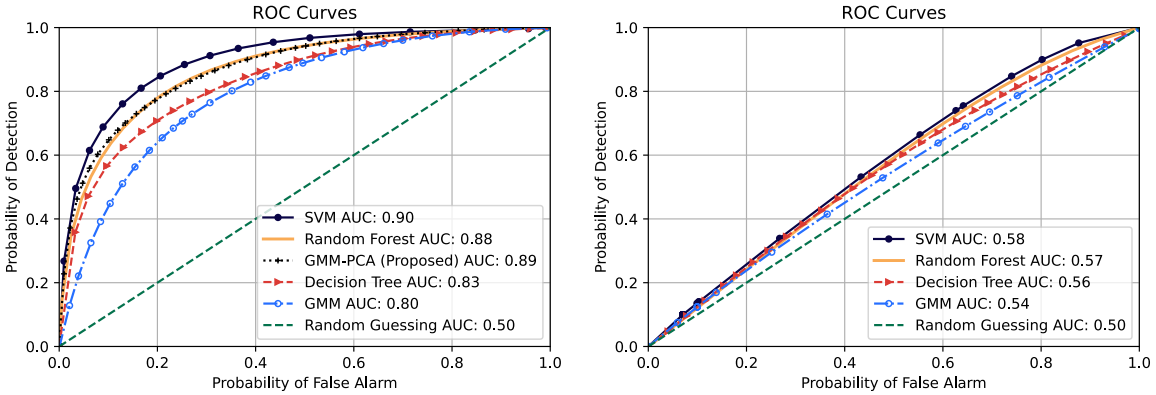
Figure 3.4: Benchmarking the detection performance of the proposed GMM-SVM approach for cooperative sensing and assessing the impact of intermittently active PUs on sensing performance.

Dimensionality Reduction for Efficient Sensing

To evaluate the effectiveness of the proposed GMM-PCA approach, we benchmark its sensing performance against supervised learning methods, including Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT). Additionally, we compare it with the standard GMM. All learning models are trained using 1200 energy vectors and tested on a separate set of 1500 held-out energy vectors. A validation set of 800 energy vectors is employed to fine-tune the hyperparameters of each learning algorithm, ensuring optimal training and classification performance. The SVM kernel is chosen to be linear to reduce the number of support vectors required for constructing the hyperplane h.

The detection performance of the proposed GMM-PCA framework is evaluated in Fig. 3.5a. Since supervised learning algorithms rely on labeled data, Fig. 3.5a clearly shows a significant performance gap between them and the GMM. Training the GMM with high-dimensional energy vectors leads to a model with poor generalization ability (overfitting), which negatively affects the overall sensing performance of the CR network. In contrast, SVM, RF, and DT are resilient to high-dimensional data and can therefore generate more effective classification models, as illustrated in Fig. 3.5a. However, by reducing the dimensionality of the training energy vectors, the proposed framework achieves performance comparable to that of RF. In Fig. 3.5b, the cooperation size n is reduced to 2 SUs. Comparing Fig. 3.5b to Fig. 3.5a reveals that increasing the number of spatially diverse SUs enhances detection performance for all learning algorithms. Furthermore, Fig. 3.5b emphasizes that using 2 SUs to sense the spectrum is not equivalent to using 25 SUs and reducing the dimensionality of their sensing data to 2 dimensions.

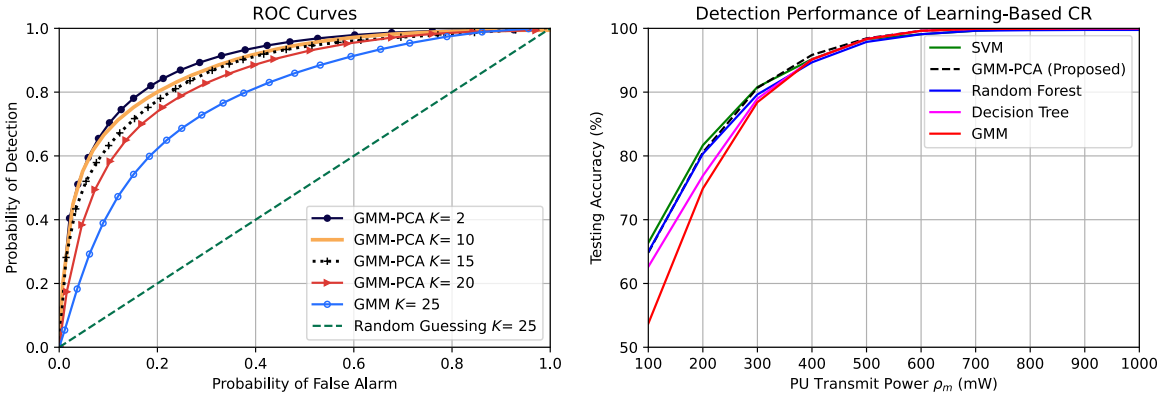
The detection performance of the proposed GMM-PCA framework is analyzed in Fig. 3.6a for different values of the number of principal components K. Fig. 3.6a illustrates that reducing the dimensionality of the training energy vectors leads to an improvement in the classification accuracy of the GMM-PCA approach. Additionally, it is evident from Fig. 3.6a that a 2-dimensional subspace provides the best performance in terms of capturing the maximum variance in the training data. PCA effectively preserves the majority of the sensing information while simultaneously reducing computational complexity, thereby enhancing the overall generalization capability of the learning



(a) ROC Curves at n= 25 SUs and m= 1 PU.

(b) ROC curves at n= 2 SUs and m= 1 PU.

Figure 3.5: Benchmarking the cooperative spectrum sensing performance of the proposed GMM-PCA against multiple learning approaches.



(a) Performance of the GMM-PCA approach for varying K with m= 1 PU.

(b) Benchmarking the testing accuracy of our proposed GMM-PCA approach against other learning approaches for varying ρ_m with $m = 1$ PU.

Figure 3.6: The effect of the number of principal components K and the PU transmit power ρ_m on the performance of the intelligent radio network.

model. As a result, the proposed intelligent radio network can establish a decision boundary with minimal classification errors, improving its detection performance and enabling the network to benefit from the performance improvements gained through cooperation.

Fig. 3.6b illustrates the classification accuracy of the proposed learning-based spectrum sensing method, comparing it with the SVM, RF, DT, and GMM frameworks under varying PU transmit power ρ_m. The accuracy is defined as the ratio of energy vectors correctly classified by the intelligent radio system to the total number of energy vectors. As ρ_m increases, the classification accuracy

improves, indicating that the ability of SUs to detect primary network activity improves. This is because, as shown in Fig. 3.7, the clusters move further apart, making them more distinguishable. From Fig. 3.6b, it can be observed that the proposed unsupervised learning method achieves classification accuracy comparable to both the SVM and RF supervised learning algorithms, without requiring labeled ground truth data or introducing communication overhead with the primary network. Additionally, the unsupervised approach demonstrates strong performance even under low SNRs.

The left plot in Fig. 3.7 shows the energy vectors projected onto a 2-dimensional subspace using PCA before training the GMM algorithm. The right plot in Fig. 3.7 presents the clustering result after training the GMM on the reduced-dimensional data. From the right plot, it is evident that the proposed GMM-PCA learning framework successfully establishes a clear decision boundary between the two clusters, H_0 and H_1, with a testing accuracy of 79.24%. Additionally, the difference between the training and testing accuracies is only 1.50%, which suggests that a well-fitting model with high generalization capability has been built. Fig. 3.8 shows the clustering results before and after training the GMM-PCA framework. Comparing Fig. 3.8 with Fig. 3.7, it is evident that as the number of PUs m increases, the clusters move further apart, making it easier for the proposed learning-based CR network to detect the presence of PUs and accurately sense channel activity.

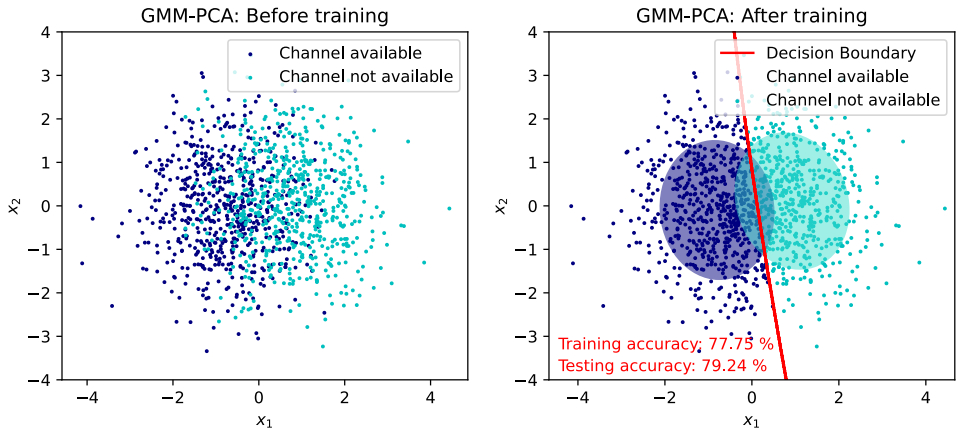


Figure 3.7: Clustering using the proposed GMM-PCA learning framework for $m = 1$ PU.

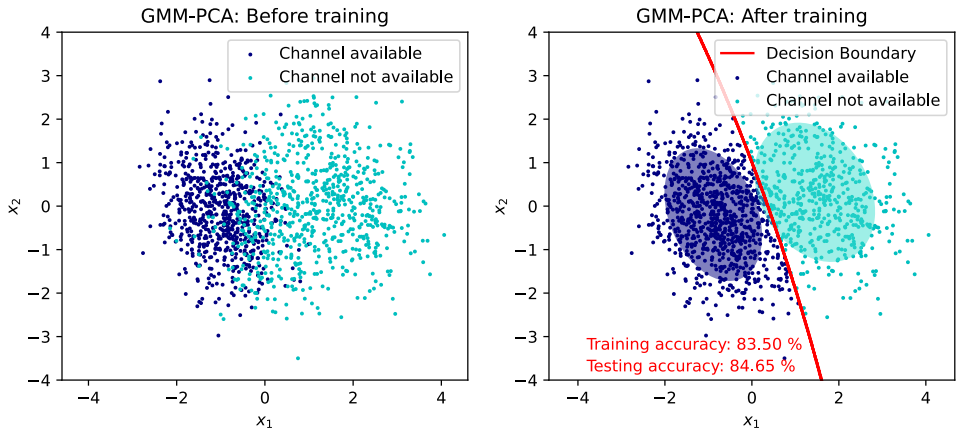


Figure 3.8: Clustering using the proposed GMM-PCA learning framework for $m = 2$ PUs.

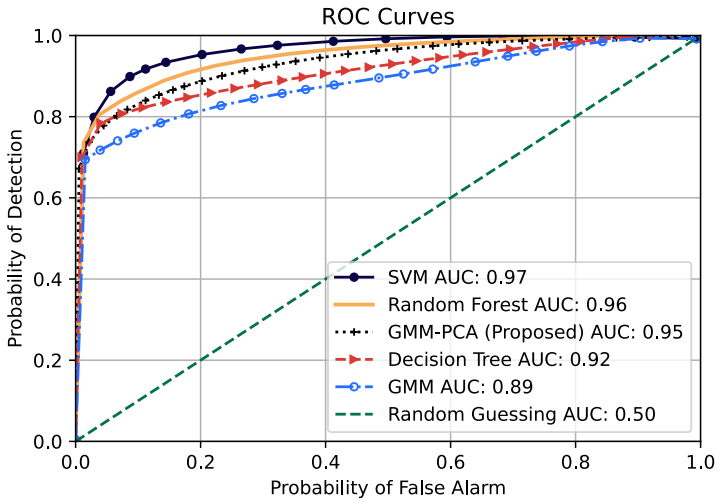


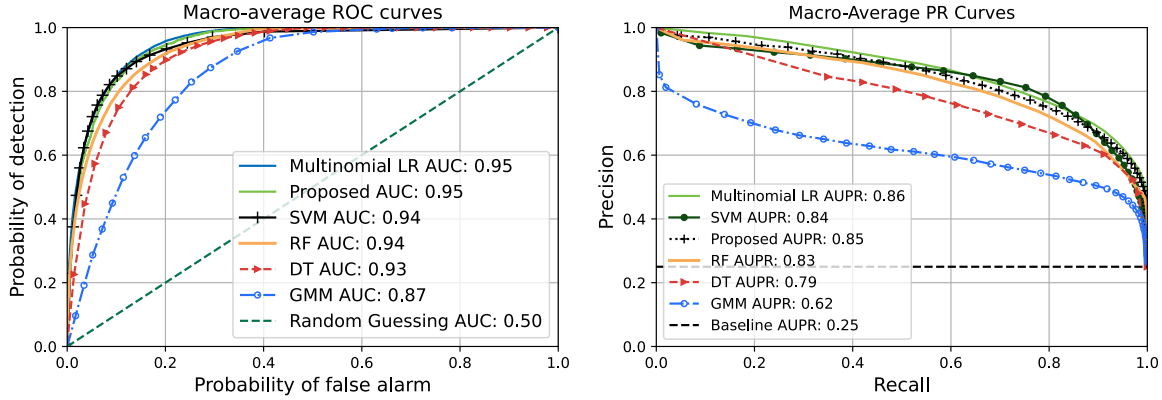
Figure 3.9: Benchmarking the detection performance of our proposed GMM-PCA approach to other learning algorithms at n= 25 SUs and $m = 2$ PUs.

Fig. 3.9 illustrates that the proposed framework significantly outperforms the GMM and DT algorithms and achieves detection performance comparable to RF. Furthermore, when comparing Fig. 3.5a and Fig. 3.9 for the same number of SUs (n= 25), it can be concluded that the ROC and AUC in Fig. 3.9 are higher than those in Fig. 3.5a. This improvement is due to the increase in the number of PUs m in the channel, which raises the energy levels in the radio environment and causes the clusters in Fig. 3.7 to separate further.

Unsupervised Learning for Situational Awareness in Hybrid CR

In our performance evaluations, we use the ROC and AUC as primary metrics. However, given that the ROC is not sensitive to data imbalance in multiclass classification, we also utilize the Precision-Recall (PR) curve and Area Under the PR Curve (AUPR) to assess predictive performance. To extend ROC and PR to multiclass prediction, we apply macro-averaging, which involves averaging C curves at each distinct x-axis point. To evaluate our unsupervised learning approach, we compare it with several supervised learning algorithms, including multinomial Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT). These supervised algorithms are trained on a balanced dataset containing 2800 energy vectors and tested on a separate set of 2400 held-out energy vectors. Additionally, we fine-tune the hyperparameters of each algorithm using a validation set consisting of 1900 energy vectors. The computations are performed on a 64-bit computer with a core i7 processor (2.8 GHz clock speed) and 16 GB of RAM.

The detection performance of the intelligent hybrid radio network is evaluated in Fig. 3.10a. Since supervised learning algorithms are trained with labeled data, Fig. 3.10a highlights the significant performance gap between these algorithms and the pure GMM. Training the GMM with high-dimensional energy vectors leads to an overfit model, which reduces the overall detection capability of the CR network. In contrast, multinomial LR, SVM, RF, and DT algorithms are robust to high-dimensional data and are therefore capable of generating more accurate prediction models. The proposed unsupervised GMM-PCA approach, however, performs comparably to both SVM and multinomial LR by projecting the energy vectors onto a two-dimensional subspace. An analysis of the PR curves in Fig. 3.10b demonstrates the impressive performance of our proposed GMM-PCA approach. In comparison to SVM, RF, and DT, our approach not only surpasses them but also achieves predictive accuracy similar to that of LR. The consistently high precision values observed with our approach emphasize its effectiveness in handling imbalanced training data. This capability is crucial, as biased learning models can negatively impact system performance. Incorrectly determining the channel state may lead to misguided decisions, potentially reducing throughput or, in worse cases, causing interference with PUs. Consequently, our proposed approach not only excels in detection performance but also plays a vital role in enhancing system reliability and reducing the



(a) Assessment of the detection performance of the CR network using the ROC curve.

(b) Assessment of the predictive capacity of the CR network using the PR curve.

Figure 3.10: Benchmarking the detection performance and the predictive capacity of the proposed GMM-PCA for hybrid CR networks against other learning algorithms.

risk of misclassification errors in hybrid networks.

Table. 3.4 presents the average training time across 100 trials for different numbers of training samples. The GMM-PCA demonstrates the shortest training time at 5000 training samples, highlighting its computational efficiency in comparison to pure GMM trained on high-dimensional data. Additionally, while both multinomial LR and GMM-PCA exhibit similar training times, GMM-PCA is trained exclusively on unlabeled data, which is easily accessible and cost-effective to gather at the FC. Therefore, our learning-based CR system effectively balances computational complexity with practicality.

The impact of the number of principal components K on the detection performance of the GMM-PCA framework is investigated in Fig. 3.11a. Reducing the dimensionality of the energy vectors

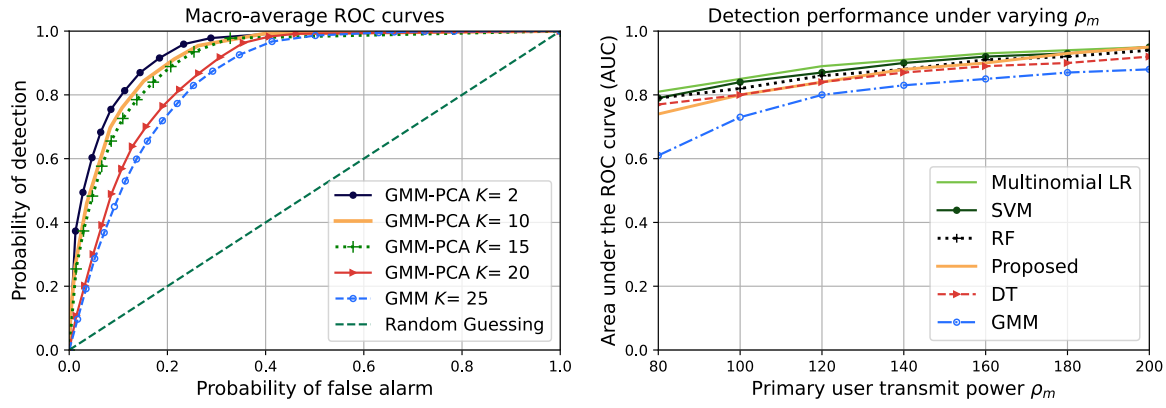
Table 3.4: Comparison of the average training time (in seconds) of the proposed GMM-PCA approach with other learning methods in hybrid CR networks.

Training Samples	**Machine Learning Algorithms Under Evaluation**					
	LR	***SVM***	***RF***	***DT***	***GMM-PCA***	***GMM***
500	0.006	0.039	0.020	0.005	0.010	0.027
1000	0.009	0.246	0.032	0.009	0.011	0.075
2000	0.013	0.640	0.050	0.016	0.014	0.153
3000	0.019	1.351	0.062	0.026	0.018	0.196
5000	0.039	3.780	0.094	0.052	0.024	0.241

improves the detection performance of the GMM-PCA. Furthermore, a two-dimensional subspace is found to be optimal for capturing the maximum variance in the training data. PCA retains most of the information gathered at the FC while reducing computational complexity and enhancing the generalization capacity of the learning model. Consequently, the proposed learning-based CR network is able to establish a decision boundary with minimal classification errors, thereby improving detection performance and enabling the network to accurately predict the primary network's channel state.

The AUC of the GMM-PCA approach is compared to multinomial LR, SVM, RF, DT, and GMM frameworks for varying PU transmit power ρ_m, with the results presented in Fig. 3.11b. It is evident that as ρ_m increases, the AUC also increases, indicating that the ability of the SUs to detect the primary network's channel state improves. This improvement is attributed to the clusters in Fig. 3.12 becoming more separable as they move farther apart. As shown in Fig. 3.11b, our proposed unsupervised learning approach achieves detection performance comparable to that of supervised learning, without requiring labeled ground truth data or incurring any communication overhead with the PUs. Additionally, our unsupervised method performs well at low SNRs in comparison to the GMM.

The clustering output before and after training the GMM-PCA framework at $\rho_m = 80$ mW is presented in Fig. 3.12. Accuracy is defined as the ratio of correct predictions to the total number of



(a) Detection performance of the proposed learning approach under varying K.

(b) Detection performance of the learning-based CR for various ρ_m.

Figure 3.11: Evaluating the detection performance of the hybrid CR network for varying K principal components and ρ_m PU transmit power.

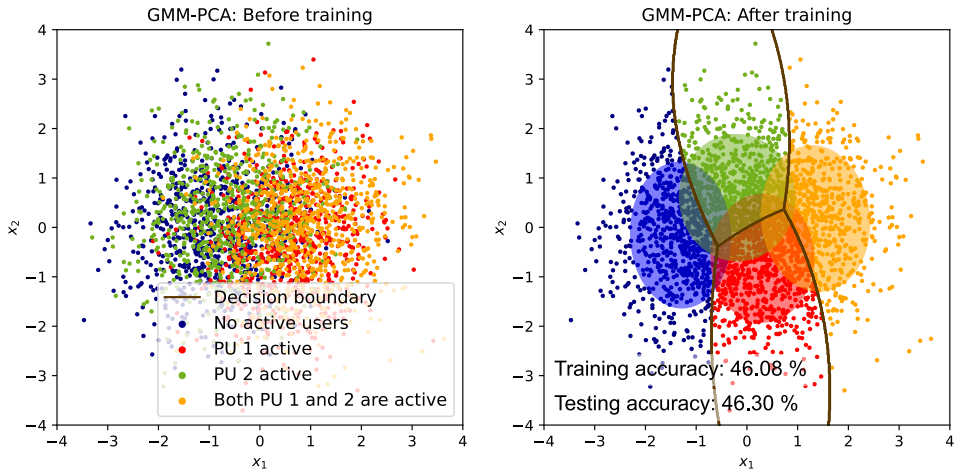


Figure 3.12: Clustering using the proposed GMM-PCA learning approach for hybrid CR networks at ρ_m= 80 mW.

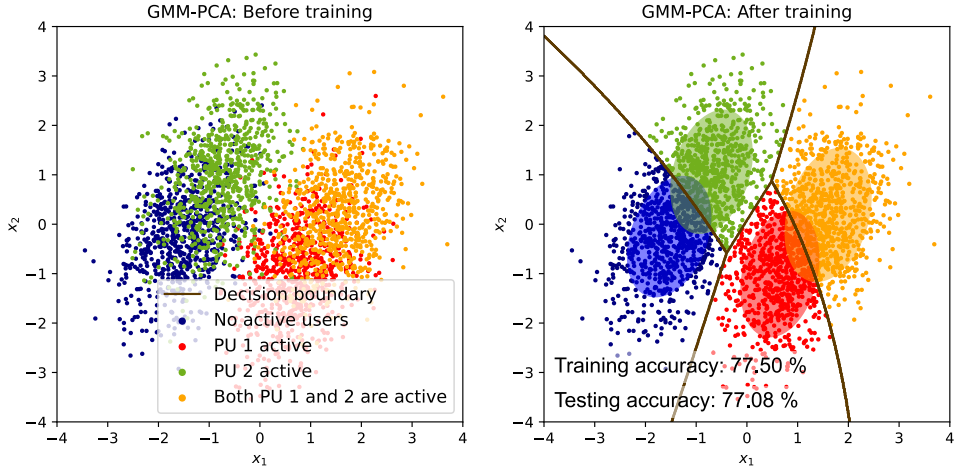


Figure 3.13: Clustering using the proposed GMM-PCA learning approach for hybrid CR at ρ_m= 200 mW.

predictions. Comparing Fig. 3.12 with Fig. 3.13, it becomes evident that as ρ_m increases from 80 mW to 200 mW, the clusters become more distinct, facilitating the hybrid CR network's ability to identify the patterns of PUs's activity. It is important to note that a dummy classifier, which always predicts the majority class, would achieve a testing accuracy of 25%. Given that the difference between the training and testing accuracy is only 0.42%, it can be concluded that the GMM-PCA method forms a learning model with a strong generalization capacity, enabling accurate and precise identification of the primary network's channel state.

3.9 Conclusions

In this chapter, we tackled the challenge of labeled data scarcity in learning-based CR networks. To address this issue, we proposed unsupervised machine learning frameworks that enhance context-awareness and ensure robust sensing performance in both interweave and hybrid interweave-underlay CR networks. Our approaches operate without relying on prior knowledge, labeled data, or cooperation between secondary and primary networks. First, we demonstrated how unsupervised learning can be utilized to generate labeled data cost-effectively, enabling the training of supervised models. Second, we illustrated how dimensionality reduction enhances both computational efficiency and the generalization capacity of unsupervised learning, ultimately improving the CR network's detection performance. Finally, we showed that our unsupervised learning approach extends to hybrid CR networks, enabling the system not only to distinguish between idle and busy channels but also to identify different activity states of the primary network. Through extensive simulations across diverse network settings, we have consistently demonstrated that our proposed unsupervised approaches achieve performance comparable to supervised learning benchmarks.

Chapter 4

Deep Representation Learning Frameworks for Advanced Spectrum Reasoning and Analysis

4.1 Introduction

Radio signals are all around us, and they play an important role in both communication and sensing as our world becomes more connected and automated. Significant efforts have been dedicated to designing and optimizing radio systems, focusing on how to represent, shape, adapt, and recover signals in challenging environments characterized by loss, nonlinearity, distortion, and interference. More recently, heavily expert-tuned functions have been substituted by feature learning using Deep Neural Networks (DNNs). The integration of Deep Learning (DL) algorithms into wireless communications has enhanced existing solutions and enabled the development of entirely new approaches, provided sufficient data is available. DL models inherently extract relevant features during training, capturing more meaningful information and enabling scalability to larger data sets while improving accuracy. In this chapter, we examine the transition from expert-designed representations to learned representations in Cognitive Radio (CR) networks. This transition aims to enhance spectrum state

identification and enable spectrum-aware networks that automatically translate channel measurements into more efficient representations, improving reasoning and analysis in a fully unsupervised manner.

4.2 Related Works

DL has notably enhanced the capabilities of CRs by applying data-driven models, particularly in intelligent monitoring and sensing domains. [63] introduced a DNN that evaluates signal energy and likelihood ratio statistics for spectrum sensing. A semi-supervised DL-based spectrum sensing approach was proposed in [53], utilizing a Variational Autoencoder (VAE) that primarily learns from unlabeled data. For non-orthogonal multiple access networks, [64] developed a Cooperative Spectrum Sensing (CSS) method that integrates both unsupervised and supervised techniques, including K-means, Gaussian Mixture Models (GMMs), and DNNs. [65] introduced an information geometry-based K-means, an unsupervised clustering technique for CSS. A supervised recurrent neural network was applied in [54] for spectrum sensing. A spectrum sensing technique combining feature extraction through autoencoders and supervised feature classification using Support Vector Machines (SVMs) was suggested in [66]. In [67], a stacked autoencoder neural network was used to preprocess raw time-domain signal samples, followed by a logistic regression classifier to detect PU transmissions.

The growing prevalence of learning-based CR operations, as highlighted by the studies above, emphasizes their critical role in enhancing network autonomy. DL-based approaches [54, 63, 66, 67] have demonstrated superior detection performance over traditional sensing methods, thanks to the neural network's ability to learn key features from signal samples. However, most DL-based techniques require a large amount of labeled data for training. In CR networks, acquiring labeled data is particularly challenging since SUs perform blind sensing without prior knowledge of the channel and cannot communicate with PUs. On the other hand, semi-supervised DL-based sensing approaches, like those in [53], only require a limited amount of labeled data. Additionally, unsupervised clustering techniques for sensing, such as those in [65], face initialization issues that may result in suboptimal outcomes or require multiple attempts to obtain a satisfactory solution.

Effective initialization necessitates prior knowledge of the sensing data, which is often unattainable. While we have previously presented unsupervised CSS algorithms in [8, 9], they rely heavily on extensive collaboration among multiple SUs to improve detection performance. However, increasing the number of SUs introduces more communication overhead between them and the Fusion Center (FC). Moreover, in some cases, only a subset of SUs may actively participate, limiting the available data features. This, in turn, reduces the degrees of freedom in the CR network and leads to a decline in overall performance.

4.3 Contributions

To address the aforementioned gaps, we introduce three representation learning frameworks to boost CSS performance in intelligent radio networks: DeepSense, DEAP learning, and G-VAP. Below, we summarize our contributions:

- We propose DeepSense which is the first fully unsupervised DL-based CSS approach for CR networks with a few cooperating SUs. DeepSense employs a Sparse Autoencoder (SAE) designed to be trained with a small amount of unlabeled low-dimensional sensing data. The SAE discovers non-linear relations between the data features in a higher-dimensional sparse space and learns a useful representation of the sensing data. A Gaussian Mixture Model (GMM) is then used to perform unsupervised clustering on the learned representations and determines the channel state.

- We enhance the DeepSense approach by introducing our award-winning DEAP learning, which also employs an SAE for representation learning but leverages the Affinity Propagation (AP) algorithm for representation clustering and spectrum state identification. Unlike traditional clustering methods, AP does not require predefined cluster centroids or prior knowledge of the number of clusters, as it infers them directly from the data. This makes DEAP learning more versatile and adaptable.

- We propose G-VAP, the first fully unsupervised deep generative approach for CSS. G-VAP employs a β-Variational Autoencoder (β-VAE) that automatically identifies independent latent

variables and encourages accurate and disentangled representations of unsupervised sensing data. Furthermore, G-VAP employs AP to cluster the learned representations and determine the channel state.

- The effectiveness of all proposed DL-based CSS approaches is rigorously evaluated through comprehensive simulations across various network settings and propagation conditions, with their performance benchmarked against leading supervised and unsupervised learning-based CSS techniques.

4.4 System Model

We consider a cognitive network comprising $n = 1, \ldots, N$ SUs and M $m = 1, \ldots, M$ PUs, all sharing a common channel with a bandwidth of ω. There is no information exchange between the SUs and PUs, and the goal is to cooperatively detect the presence or absence of PUs. The channel between the m-th PU and the n-th SU is denoted as $h_{m,n}$. To model $h_{m,n}$, we adopt a Nakagami-ν distribution, which effectively represents both indoor and outdoor multipath fading channels. The Probability Density Function (PDF) of the Nakagami-ν distribution is given by

$$f_{h_{m,n}}(\gamma) = \frac{2}{\Gamma(\nu)}(\frac{\nu}{\overline{\gamma}})^{\nu}\gamma^{2\nu-1}\exp\left(-\frac{\nu\gamma}{\overline{\gamma}}\right), \quad \gamma > 0, \quad \nu > 0. \tag{4.1}$$

Here, $\nu \geq 0.5$ represents the Nakagami multipath fading parameter, which quantifies the severity of fading. $\overline{\gamma}$ denotes the average received Signal-to-Noise Ratio (SNR), and $\Gamma(.)$ is the Gamma function. The Fusion Center (FC) can either be one of the SUs or an additional node with an external connection, such as a cluster head or a base station. Each SU utilizes an energy detector to measure the channel's energy levels and transmits these measurements to the FC. The FC, in turn, applies a DL algorithm to analyze the spectrum data and identify the spectrum state. The SUs conduct energy measurements over a duration of τ seconds. The licensed activity detection problem can be

formulated as a binary hypothesis test

$$H_0 : E_n(i) = N_n(i), \tag{4.2}$$

$$H_1 : E_n(i) = \sum_{m=1}^{M} s_m h_{m,n} X_m(i) + N_n(i). \tag{4.3}$$

Here, $E_n(i)$ is the i-th channel measurement taken by the n-th SU. s_m denotes the activity status of the m-th PU, where $s_m = 1$ indicates that the PU is actively using the channel, and $s_m = 0$ means it is inactive. We adopt a generalized PU model in which multiple PUs transition between active and inactive states. The channel is considered unavailable to the CR network if at least one PU is active ($s_m = 1$ for some m). It is deemed available only when all PUs are inactive ($s_m = 0$ for all m). The unknown transmitted signal from the m-th PU is represented as $X_m(i)$. No prior knowledge of the PUs transmit power or the prior probability of each hypothesis is assumed. The thermal noise $N_n(i)$ follows a Gaussian distribution $\mathcal{N}(0, \sigma_n^2)$, where $\sigma_n^2 = \mathbb{E}[|N_n(i)|^2]$ represents the noise's Power Spectral Density (PSD). Consequently, the spectrum energy level at the n-th SU, normalized by σ_n^2, is given by

$$y_n = \frac{2}{\sigma_n^2} \sum_{i=1}^{\omega\tau} |E_n(i)|^2, \tag{4.4}$$

where y_n follows a non-central chi-squared distribution with $q = 2\omega\tau$ degrees of freedom and a non-centrality parameter ζ_n as

$$\zeta_n = \frac{2\tau}{\sigma_n^2} \sum_{m=1}^{M} s_m g_{m,n} \rho_m. \tag{4.5}$$

The power attenuation from the m-th PU to the n-th SU is

$$g_{m,n} = |h_{m,n}|^2 = D_{m,n}^{-\alpha} \cdot \psi_{m,n} \cdot \nu_{m,n}, \tag{4.6}$$

where D is the Euclidean distance, α is the path loss exponent, $\psi_{m,n}$ is the shadow fading component, and $\nu_{m,n}$ is the multipath fading component. Finally, the m-th PU's transmit power is

$$\rho_m = \frac{\sum_{i=1}^{w\tau} \mathbb{E}[|X_m(i)|^2]}{\tau}. \tag{4.7}$$

Let $\mathbf{S}$ represent the activity state vector that captures the activity of the M PUs, such that $\mathbf{S} = (s_1, \ldots, s_M)$. Assume that each SU collects $\omega\tau$ signal samples during a sensing interval. When $\omega\tau$ is large, the central limit theorem suggests that the energy level y_n reported by the n-th SU can be approximated by a Gaussian distribution $\mathcal{N}(\mu_{y_n|\mathbf{S}=\mathbf{s}}, \sigma^2_{y_n|\mathbf{S}=\mathbf{s}})$, with the following parameters

$$\mu_{y_n|\mathbf{S}=\mathbf{s}} = \mathbb{E}[y_n|\mathbf{S}=\mathbf{s}] = 2\omega\tau + \frac{2\tau}{\sigma_n^2}\sum_{m=1}^{M} s_m g_{m,n}\rho_m, \tag{4.8}$$

$$\sigma^2_{y_n|\mathbf{S}=\mathbf{s}} = \mathbb{E}[(y_n - \mu_{y_n|\mathbf{S}=\mathbf{s}})^2|\mathbf{S}=\mathbf{s}] = 4\omega\tau + \frac{8\tau}{\sigma_n^2}\sum_{m=1}^{M} s_m g_{m,n}\rho_m. \tag{4.9}$$

It is important to note that the local users do not make any decisions; instead, the measured energy levels are directly transmitted to the FC. The FC is the sole entity responsible for making decisions based on the collected measurements from various users. It aggregates all the energy levels from the N SUs to form an energy vector $\mathbf{y} = (y_1, ..., y_N) \in \mathbb{R}^N$. As a result, the distribution of $\mathbf{y}$, conditioned on the current activity state vector $\mathbf{s}$, follows a multivariate Gaussian distribution characterized by

$$\boldsymbol{\mu}_{\mathbf{y}|\mathbf{S}=\mathbf{s}} = (\mu_{y_1|\mathbf{S}=\mathbf{s}}, ..., \mu_{y_N|\mathbf{S}=\mathbf{s}}) \tag{4.10}$$

$$\boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{S}=\mathbf{s}} = diag(\sigma^2_{y_1|\mathbf{S}=\mathbf{s}}, ..., \sigma^2_{y_N|\mathbf{S}=\mathbf{s}}). \tag{4.11}$$

$diag(\mathbf{v})$ creates a diagonal square matrix where the elements of the vector $\mathbf{v}$ are positioned along its principal diagonal.

To construct a learning model with a good generalization capacity, the first step is to collect a sufficient number of training energy vectors. Let $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_L\}$ be the set L collected energy vectors at the FC. Our goal is to utilize this data exclusively to learn an efficient representation that improves the detection performance of the CR network. In other words, we use the raw collected energy vectors without their corresponding channel state labels (H_0/H_1) to develop a learning model capable of automatically determining the spectrum state. In the following sections, we introduce three novel unsupervised representation learning frameworks designed to help the intelligent radio network uncover hidden features and nonlinear patterns, enabling more effective channel state

identification.

4.5 DeepSense

Each cooperating SU measures the spectrum energy and sends the data to the FC, which constructs an energy vector $\mathbf{y}$ based on the reported energy levels. As the number of cooperating SUs N increases, the dimensionality of the energy vectors also increases. However, when only a small number of SUs are active, the energy vectors at the FC become low-dimensional, which significantly reduces the degrees of freedom in the CR network and hampers its performance. To enhance the detection capability of the CR network with a limited number of cooperating SUs, we utilize an overcomplete autoencoder that maps the energy vectors to a higher-dimensional latent space. This allows the autoencoder to learn meaningful representations of the sensing data and identify non-linear relationships between the energy levels. As a result, the FC can better cluster the sensing data, improving the overall PU detection performance.

4.5.1 Representation Learning Using Sparse Autoencoders

An autoencoder is a neural network comprising two components: an encoder and a decoder. We use a fully connected Deep Neural Network (DNN) for both parts, as shown in Fig. 4.1. The input layer of the autoencoder contains N neurons corresponding to the number of actively cooperating SUs, with each input energy vector $\mathbf{y} = (y_1, \ldots, y_N) \in \mathbb{R}^N$. The encoder has one hidden layer with J neurons, where $N < J$. The output layer of the encoder consists of K neurons, where $N < J < K$. The latent representation output of the encoder can be expressed as $\mathbf{z} = f_{\theta_{\text{enc}}}(\mathbf{y}) \in \mathbb{R}^K$, where $\theta_{\text{enc}} = \{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}_{i=\{1,2\}}$ denotes the weight matrices and vector biases of the encoder network. On the other hand, the decoder's goal is to reconstruct the original data from the latent representation as follows $\hat{\mathbf{y}} = g_{\theta_{\text{dec}}}(\mathbf{z})$, where $\theta_{\text{dec}} = \{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}_{i=\{3,4\}}$ denotes the weight matrices and vector biases of the decoder network. Thus, the encoder and decoder outputs can be

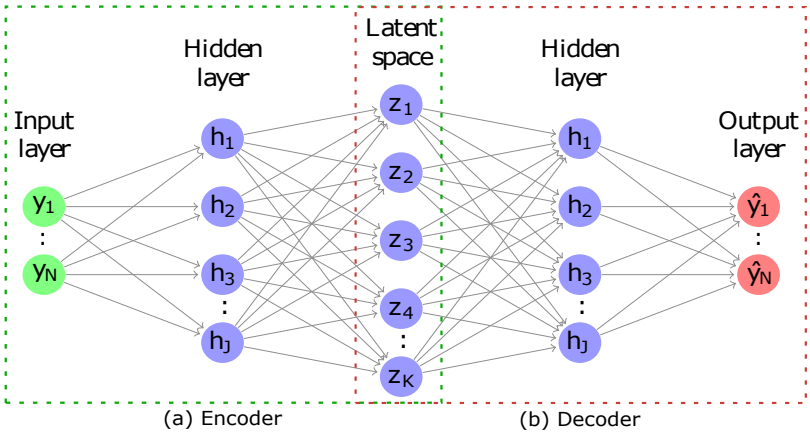


Figure 4.1: The architecture of the sparse autoencoder for representation learning in the DeepSense approach.

written as

$$f_{\theta_{\text{enc}}}(\mathbf{y}) = \mathbf{z} = \sigma_{\tanh}\Big(\sigma_{\tanh}(\mathbf{y}\mathbf{W}^{(1)} + \mathbf{b}^{(1)})\mathbf{W}^{(2)} + \mathbf{b}^{(2)}\Big), \tag{4.12}$$

$$g_{\theta_{\text{dec}}}(\mathbf{z}) = \hat{\mathbf{y}} = \sigma_{\tanh}\Big(\sigma_{\tanh}(\mathbf{z}\mathbf{W}^{(3)} + \mathbf{b}^{(3)})\mathbf{W}^{(4)} + \mathbf{b}^{(4)}\Big), \tag{4.13}$$

where the superscript of $\mathbf{W}$ and $\mathbf{b}$ represents the layer number of the autoencoder. The activation function $\sigma_{\tanh}$ represents the element-wise application of the hyperbolic tangent function. The $\tanh$ function is chosen, since it is centered around zero, leading to faster convergence time. Additionally, the $\tanh$ function introduces non-linearity into the model, enabling the SAE to learn non-linear relationships between the energy levels reported by the SUs in the higher-dimensional sparse feature space. This non-linearity is essential for modeling complex patterns in the sensing data, enhancing the CR network's ability to detect and classify primary user activity accurately.

Since an autoencoder aims to replicate its inputs at the output, it is essentially addressing a regression problem. As a result, we use the Mean Squared Error (MSE) loss to measure the reconstruction error during the training process, which is defined as

$$\begin{aligned}\mathcal{L}(\theta_{\text{enc}}, \theta_{\text{dec}}; \mathbf{Y}) &= \frac{1}{T}\sum_{t=1}^{T} \|\mathbf{y}_t - \hat{\mathbf{y}}_t\|^2 \\ &= \frac{1}{T}\sum_{t=1}^{T} \left\|\left(\mathbf{y}_t - g_{\theta_{\text{dec}}}(f_{\theta_{\text{enc}}}(\mathbf{y}_t))\right)\right\|^2.\end{aligned} \tag{4.14}$$

$\|.\|$ is the L2 norm. $\mathbf{Y}$ is a matrix $T \times N$ which represents a batch of T energy vectors.

Let $z_k(\mathbf{y}_t)$ represent the output activation of the k-th neuron in the encoder's output layer (latent space) for a given input energy vector $\mathbf{y}_t$. The average activation of the k-th neuron over a batch of T training energy vectors is then

$$\hat{\gamma}_k = \frac{1}{T} \sum_{t=1}^{T} \Big[z_k(\mathbf{y}_t) \Big]. \tag{4.15}$$

By copying the inputs to the outputs, the autoencoder essentially learns an identity function. However, by imposing constraints on the network, it is possible to uncover interesting structures within the sensing data. To achieve this, we introduce a sparsity penalty during training, ensuring that the encoder's output neurons remain inactive most of the time. Specifically, we aim to constrain $\hat{\gamma}_k = \gamma$, where γ represents the sparsity parameter, typically set to a value close to -1. As a result, the training loss function $\mathcal{L}$ in (4.14) becomes

$$\mathcal{L}_{\text{sparse}}(\theta_{\text{enc}}, \theta_{\text{dec}}; \mathbf{Y}) = \mathcal{L} + \beta \sum_{k=1}^{K} \text{KL}(\gamma || \hat{\gamma}_k), \tag{4.16}$$

where $\text{KL}(\gamma||\hat{\gamma}_k) = \gamma \log \frac{\gamma}{\hat{\gamma}_k} + (1-\gamma) \log \frac{1-\gamma}{1-\hat{\gamma}_k}$ is the Kullback-Leiber (KL) divergence between two Bernoulli random variables whose means are γ and $\hat{\gamma}_k$, respectively. β controls the weight of the sparsity term. If $\hat{\gamma}_k = \gamma$ then $\text{KL}(\hat{\gamma}_k||\gamma)$ = 0, otherwise $\text{KL}(\hat{\gamma}_k||\gamma)$ increases monotonically as $\hat{\gamma}_k$ diverges from γ.

During training, the goal is to optimize $\{\theta_{\text{enc}}, \theta_{\text{dec}}\}$ such that the loss in (4.16) is minimum. That is,

$$\theta^*_{\text{enc}}, \theta^*_{\text{dec}} = \arg \min_{\theta_{\text{enc}}, \theta_{\text{dec}}} \mathcal{L}_{\text{sparse}}(\theta_{\text{enc}}, \theta_{\text{dec}}; \mathbf{Y}). \tag{4.17}$$

The Backpropagation (BP) algorithm [68] is used to efficiently compute the gradient of the loss in (4.16) with respect to w.r.t $\theta_{\text{enc}} = \{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}_{i=\{1,2\}}$ and $\theta_{\text{dec}} = \{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}_{i=\{3,4\}}$. Considering

a single training energy vector $\mathbf{y}$, the gradient of the loss w.r.t weights of the SAE is as follows

$$\begin{aligned}
\nabla_{\mathbf{w}^{(1)}}\mathcal{L} &= \frac{\partial \mathcal{L}_{\text{sparse}}}{\partial g_{\theta_{\text{dec}}}(f_{\theta_{\text{enc}}}(\mathbf{y}))} \cdot \frac{\partial g_{\theta_{\text{dec}}}(f_{\theta_{\text{enc}}}(\mathbf{y}))}{\partial f_{\theta_{\text{enc}}}(\mathbf{y})} \cdot \frac{\partial f_{\theta_{\text{enc}}}(\mathbf{y})}{\partial \mathbf{h}^{(2)}} \cdot \frac{\partial \mathbf{h}^{(2)}}{\partial \mathbf{w}^{(1)}}, \\
\nabla_{\mathbf{w}^{(2)}}\mathcal{L} &= \frac{\partial \mathcal{L}_{\text{sparse}}}{\partial g_{\theta_{\text{dec}}}(f_{\theta_{\text{enc}}}(\mathbf{y}))} \cdot \frac{\partial g_{\theta_{\text{dec}}}(f_{\theta_{\text{enc}}}(\mathbf{y}))}{\partial f_{\theta_{\text{enc}}}(\mathbf{y})} \cdot \frac{\partial f_{\theta_{\text{enc}}}(\mathbf{y})}{\partial \mathbf{w}^{(2)}}, \\
\nabla_{\mathbf{w}^{(3)}}\mathcal{L} &= \frac{\partial \mathcal{L}_{\text{sparse}}}{\partial g_{\theta_{\text{dec}}}(f_{\theta_{\text{enc}}}(\mathbf{y}))} \cdot \frac{\partial g_{\theta_{\text{dec}}}(f_{\theta_{\text{enc}}}(\mathbf{y}))}{\partial \mathbf{h}^{(3)}} \cdot \frac{\partial \mathbf{h}^{(3)}}{\partial \mathbf{w}^{(3)}}, \\
\nabla_{\mathbf{w}^{(4)}}\mathcal{L} &= \frac{\partial \mathcal{L}_{\text{sparse}}}{\partial g_{\theta_{\text{dec}}}(f_{\theta_{\text{enc}}}(\mathbf{y}))} \cdot \frac{\partial g_{\theta_{\text{dec}}}(f_{\theta_{\text{enc}}}(\mathbf{y}))}{\partial \mathbf{w}^{(4)}}.
\end{aligned} \tag{4.18}$$

Similarly, the gradient of the loss in (4.16) w.r.t to the bias vectors $\{\nabla_{\mathbf{b}^{(i)}}\mathcal{L}\}_{i=1,\dots,4}$ can be obtained by applying the chain rule.

Stochastic Gradient Descent (SGD) is then employed to iteratively adjust the parameters of the SAE by calculating the gradient of the loss with respect to the weight matrices and bias vectors, which are derived using the BP algorithm. Thus, the parameter update of the SAE for a batch of energy vectors $\mathbf{Y}$ at each iteration is given by

$$\theta^{*}_{\text{enc}} := \theta_{\text{enc}} - \eta \nabla_{\theta_{\text{enc}}} \mathcal{L}_{\text{sparse}}(\theta_{\text{enc}}, \theta_{\text{dec}}; \mathbf{Y}), \tag{4.19}$$

$$\theta^{*}_{\text{dec}} := \theta_{\text{dec}} - \eta \nabla_{\theta_{\text{dec}}} \mathcal{L}_{\text{sparse}}(\theta_{\text{enc}}, \theta_{\text{dec}}; \mathbf{Y}), \tag{4.20}$$

where η is the learning rate that determines the step size during each iteration of SGD.

4.5.2 Gaussian Mixture Models for Unsupervised Clustering

Let $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_L\}$ be the set of L transformed energy vectors using the encoder network of the proposed SAE, where $\mathbf{z}_l \in \mathbb{R}^K$ and $K > N$. The FC focuses on distinguishing between two main clusters: channel available H_0 and channel unavailable H_1. Therefore, we train an unsupervised Gaussian Mixture Model (GMM) to cluster the sensing data in the latent sparse space. The parameters of the Gaussian densities $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}|\mathbf{S}=\mathbf{0}}, \boldsymbol{\Sigma}_{\mathbf{z}|\mathbf{S}=\mathbf{0}})$ and $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}|\mathbf{S}=\mathbf{s}}, \boldsymbol{\Sigma}_{\mathbf{z}|\mathbf{S}=\mathbf{s}})$, corresponding to clusters H_0 and H_1 respectively, are unknown prior to training. Furthermore, the mixing weights v_1 and v_2 are unknown. The Expectation-Maximization (EM) algorithm [58] is therefore used to estimate the collection of unknown parameters $\boldsymbol{\theta}$ that maximizes the log-likelihood of $\mathbf{Z}$ [58]. The EM algorithm

was detailed in Section 3.6.2.

Let $\mathbf{z}^* \in \mathbb{R}^K$ be a testing feature vector, the log-likelihood of $\mathbf{z}^*$ using the estimated parameters $\boldsymbol{\theta}^*$ is

$$\omega(\mathbf{z}^*|\boldsymbol{\theta}^*) = \ln\left(v_2^*.\phi(\mathbf{z}^*|\boldsymbol{\mu_2^*}, \boldsymbol{\Sigma_2^*})\right) - \ln(v_1^*.\phi(\mathbf{z}^*|\boldsymbol{\mu_1^*}, \boldsymbol{\Sigma_1^*})). \tag{4.21}$$

$\ln\left(v_2^*.\phi(\mathbf{z}^*|\boldsymbol{\mu_2}^*, \boldsymbol{\Sigma_2}^*)\right)$ is the log-likelihood that $\mathbf{z}^*$ belongs to H_1. Similarly, $\ln\left(v_1^*.\phi(\mathbf{z}^*|\boldsymbol{\mu_1}^*, \boldsymbol{\Sigma_1}^*)\right)$ is the log-likelihood that $\mathbf{z}^*$ belongs to H_0. For a decision threshold of δ, if $\omega(\mathbf{z}^*|\boldsymbol{\theta}^*) \geq \delta$, then the channel state is H_1, otherwise it is H_0. The probability of false alarm $P_{fa} = P(\omega(\mathbf{z}^*|\boldsymbol{\theta}^*) \geq \delta|H_0)$ can be decreased at the expense of the probability of detection $P_d = P(\omega(\mathbf{z}^*|\boldsymbol{\theta}^*) \geq \delta|H_1)$ by increasing δ since $\mathbf{z}^*$ is more likely to be classified as H_0 if the value of δ is high.

4.6 DEAP Learning

The DeepSense approach has potential in enhancing data clustering through the use of a Sparse Autoencoder (SAE), mapping sensing data into a high-dimensional sparse latent space. While this method improves clustering at the FC, it inherits limitations from its reliance on the GMM. Due to the changing nature of the radio environment, a GMM faces challenges with cluster centroid initialization and requires predefining the number of clusters expected in the data. This often leads to multiple runs to achieve a satisfactory solution, which is not effective and is computationally expensive. Additionally, the need for optimal initialization implies that prior knowledge about the sensing data is required, which is not always feasible to obtain. To address these challenges, we propose DEAP learning, which also capitalizes on representation learning by an SAE but instead leverages the Affinity Propagation (AP) algorithm for clustering the latent representations of the sensing data to determine the channel state (H_0/H_1).

4.6.1 Affinity Propagation: Unsupervised Clustering Based on Message-Passing

AP clusters data based on similarity measures. The algorithm takes as input a set of real-valued similarities between data points. Let $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_L\}$ represent the set of L transformed energy vectors obtained from the trained encoder network shown in Fig. 4.1, where each $\mathbf{z}_l \in \mathbb{R}^K$ with $K > N$. The similarity between points in $\mathbf{Z}$ is defined as the negative Euclidean distance, given by

$u(l,i) = -||\mathbf{z}_l - \mathbf{z}_i||$ for $\{l,i\} = 1,\ldots,L$ and $l \neq i$. Unlike K-means or GMMs, which require a predefined number of clusters, AP employs a "preference" parameter to determine which data points are more likely to be chosen as "exemplars" or cluster centers. Since the CR network lacks prior knowledge about the sensing data, we assume all points have an equal likelihood of being exemplars and set the preference to a common value.

During the training phase of the AP algorithm, two types of messages are exchanged between the data points in $\mathbf{Z}$. The first type, called "responsibility", is transmitted from a point l to a candidate exemplar i. The value of $r(l,i)$ quantifies how well point i serves as an exemplar for point l, given all potential exemplars for l. The second type, termed "availability", is sent from a candidate exemplar i to point l. The value of $a(l,i)$ indicates the appropriateness of point l selecting point i as its exemplar, considering the overall support for l from other points. The update rules for $r_t(l,i)$ and $a_t(l,i)$ are given as follows

$$r_t(l,i) \longleftarrow u(l,i) - \max_{i's.t.i' \neq i} \{a(l,i') + u(l,i')\}, \tag{4.22}$$

$$a_t(l,i) \longleftarrow \min\left\{0, r(i,i) + \sum_{l's.t.l' \notin \{l,i\}} \max\{0, r(l',i)\}\right\}. \tag{4.23}$$

Both $r_t(l,i)$ and $a_t(l,i)$ are set to zero at the start of training and updated iteratively until convergence. The "self-availability" is updated differently

$$a_t(i,i) \leftarrow \sum_{l's.t.l' \neq i} \max\{0, r(l',i)\}. \tag{4.24}$$

This message represents the gathered evidence indicating that point i serves as an exemplar, based on the positive responsibilities received from other points considering it as a candidate exemplar.

To stabilize numerical oscillations during training, a damping factor λ is utilized. As a result, the updates of r and a are described as

$$r_{t+1}(l,i) = \lambda \cdot r_t(l,i) + (1-\lambda) r_{t+1}(l,i), \tag{4.25}$$

$$a_{t+1}(l,i) = \lambda \cdot a_t(l,i) + (1-\lambda) a_{t+1}(l,i). \tag{4.26}$$

Here, t denotes the iteration step. At each iteration, a point i is designated as an exemplar if $a(i,i) + r(i,i) > 0$. Additionally, AP assigns cluster centroids based on $\arg\max_i [a(l,i) + r(l,i)]$ for $l = 1, ..., L$. The update rules in (4.23) involve straightforward, local computations that are easy to implement, requiring message exchanges only between pairs of points with known similarities. Moreover, these computations can be significantly accelerated using hardware such as Graphical Processing Units (GPUs) or Tensor Processing Units (TPUs).

The convergence criterion used in this work is based on monitoring whether local decisions remain unchanged after a predefined number of iterations. Once this condition is met, the algorithm is considered to have converged. Let $\mathbf{z}_0$ and $\mathbf{z}_1$ represent the cluster exemplars corresponding to H_0 and H_1, respectively. The similarity between a test vector $\mathbf{z}^*$ and $\mathbf{z}_0$ is given by u_0, while the similarity between $\mathbf{z}^*$ and $\mathbf{z}_1$ is denoted as u_1. For a given threshold δ, if $u_0 - u_1 > \delta$, then $\mathbf{z}^*$ is assigned to cluster H_0; otherwise, it is classified as belonging to H_1. Increasing δ reduces the probability of misdetection but at the expense of a higher false alarm probability, as a larger δ increases the likelihood of $\mathbf{z}^*$ being classified into cluster H_1.

4.7 G-VAP

In the previous sections, we introduced DeepSense and DEAP learning, both designed for CR networks with limited cooperating SUs, which results in reduced degrees of freedom. However, another challenge arises when dealing with a larger pool of cooperating SUs. While increased cooperation can enhance sensing capabilities, we have previously proved that it also introduces complexities for unsupervised models due to the high-dimensional nature of the data [8]. To address this, our goal is to develop an unsupervised deep representation learning model that learns accurate, separable, and efficient representations of high-dimensional sensing data. This, in turn, enables unsupervised clustering algorithms to better distinguish patterns and determine the channel state with greater precision. To achieve this, we propose G-VAP, the first fully unsupervised deep generative approach for CSS. G-VAP employs a β-Variational Autoencoder (β-VAE) that automatically identifies independent latent variables and encourages accurate and disentangled representations of unsupervised sensing data. Additionally, it incorporates the AP algorithm to

cluster the learned representations, effectively determining whether a channel is vacant or occupied.

4.7.1 β-VAE: Joint Deep Generative Modeling and Representation Learning

Let $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_L\}$ be the set L collected energy vectors at the FC. Our objective is for each observed data point $\mathbf{y}_l$ to be associated with a corresponding low-dimensional latent variable $\mathbf{z}_l$. To do so, we are primarily interested in two tasks: (1) For a fixed set of model parameters θ, for each $\mathbf{y}_l$, compute the posterior distribution $p_\theta(\mathbf{z}_l|\mathbf{y}_l)$; (2) Maximize the likelihood of the observed data under θ. However, solving the posterior $p_\theta(\mathbf{z}_l|\mathbf{y}_l)$ using Bayes' theorem is intractable due to the fact that the denominator requires marginalizing over $\mathbf{z}_l$ as

$$p_\theta(\mathbf{z}_l|\mathbf{y}_l) = \frac{p_\theta(\mathbf{y}_l|\mathbf{z}_l)p(\mathbf{z}_l)}{\int p_\theta(\mathbf{y}_l|\mathbf{z}_l)p(\mathbf{z}_l)d\mathbf{z}_l}. \tag{4.27}$$

This marginalization requires solving an integral over all of the dimensions of the latent space, which is not feasible to calculate. Estimating θ via maximum likelihood estimation also requires solving the following integral

$$\begin{aligned} \hat{\theta} &:= \arg\max_\theta \prod_{l=1}^{L} p_\theta(\mathbf{y}_l) \\ &= \arg\max_\theta \prod_{l=1}^{L} \int p_\theta(\mathbf{y}_l|\mathbf{z}_l)p(\mathbf{z}_l)d\mathbf{z}_l. \end{aligned} \tag{4.28}$$

Variational Autoencoders (VAEs) find approximate solutions to the intractable inference problems (4.27) and (4.28) by relying on variational Bayesian inference to learn a latent-space representation of the sensing data. A VAE is a deep probabilistic model that consists of two main structures: the encoder (inference model) and the decoder (generative model). The goal is to maximize the likelihood of the observed data under the generative model (decoder) $p_\theta(\mathbf{y}|\mathbf{z})p(\mathbf{z})$, effectively learning the best set of parameters θ that explain the data distribution. $p(\mathbf{z})$ is the prior on the latent space, which we consider to be the centered isotropic multivariate Gaussian $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This prior encourages disentanglement of the posterior $q_\phi(\mathbf{z}|\mathbf{y})$ and regulates the latent space capacity.

Given the intractability of $p_\theta(\mathbf{z}|\mathbf{y})$, VAEs approximate this posterior with $q_\phi(\mathbf{z}|\mathbf{y})$, which is parameterized by the encoder network. That is, during training both ϕ and θ are iteratively optimized

such that $q_\phi(\mathbf{z}|\mathbf{y}) \approx p_\theta(\mathbf{z}|\mathbf{y})$. The goal of the encoder is to learn a function that maps an input energy vector $\mathbf{y}$ to a low-dimensional latent Gaussian variable $\mathbf{z}$ characterized by the parameters μ_ϕ and σ_ϕ. That is,

$$\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{y}) = \mathcal{N}(\mathbf{z}; \mu_\phi, \sigma_\phi^2 \mathbf{I}). \tag{4.29}$$

Fig. 4.2 shows the graphical representation of the VAE model. Solid lines indicate the generative process $p_\theta(\mathbf{y}|\mathbf{z})p(\mathbf{z})$, while dashed lines represent the variational approximation $q_\phi(\mathbf{z}|\mathbf{y})$ to the intractable posterior $p_\theta(\mathbf{z}|\mathbf{y})$.

The proposed VAE architecture is shown in Fig. 4.3. The encoder employs a Deep Neural Network (DNN) with parameters $\phi = \{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}_{i=1,2}$, where $\mathbf{W}^{(i)}$ and $\mathbf{b}^{(i)}$ denote the weight matrices and bias vectors, respectively, for each layer i. The encoder's input layer consists of N neurons, corresponding to the number of energy levels collected by the cooperative SUs. The architecture includes one hidden layer with J neurons, where $J < N$, and an output layer, which represents the latent space, containing K neurons such that $K < J < N$. The encoder's output can be mathematically expressed as

$$f_\phi(\mathbf{y}) = \mu_\phi, \sigma_\phi = \Big(\tanh(\mathbf{y}\mathbf{W}^{(1)} + \mathbf{b}^{(1)}) \Big) \mathbf{W}^{(2)} + \mathbf{b}^{(2)}. \tag{4.30}$$

In the proposed VAE, the $\tanh(\cdot)$ function is used to capture non-linear relationships in the energy

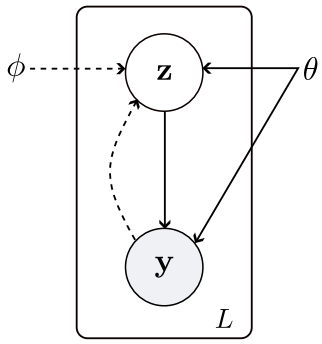


Figure 4.2: Graphical representation of the VAE model. The encoder's variational parameters ϕ are learned alongside the decoder's generative parameters θ during training.

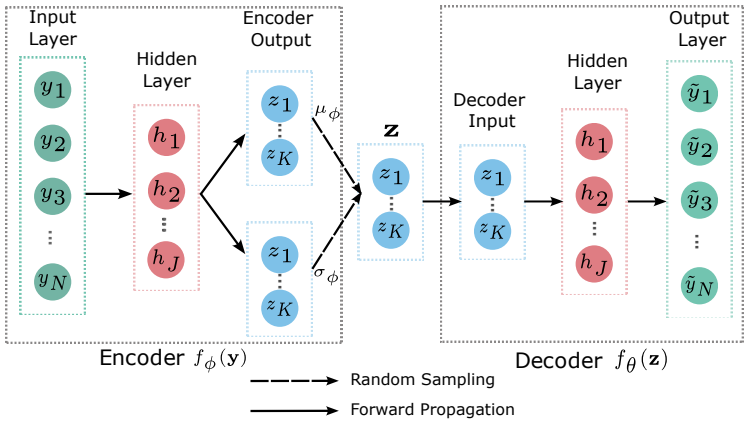


Figure 4.3: The architecture of the β-variational autoencoder for representation learning in the proposed G-VAP approach.

levels gathered by the SUs. The decoder is a DNN with three layers containing K, J, and N neurons, respectively. The parameters of the decoder are $\theta = \{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}_{i=3,4}$. The decoder's role is to reconstruct an energy vector $\hat{\mathbf{y}}$ from the latent variable $\mathbf{z}$. The decoder's output $\hat{\mathbf{y}}$ can be expressed as

$$f_\theta(\mathbf{z}) = \hat{\mathbf{y}} = \sigma_{\text{sigmoid}}\Big(\tanh(\mathbf{z}\mathbf{W}^{(3)} + \mathbf{b}^{(3)})\Big)\mathbf{W}^{(4)} + \mathbf{b}^{(4)}, \tag{4.31}$$

where $\sigma_{\text{sigmoid}}(x)$ is the sigmoid function defined as $\sigma_{\text{sigmoid}}(x) = 1/(1+e^{-x})$.

We employ an advanced variant of VAEs known as β-VAE. This model introduces a hyperparameter β to regulate reconstruction accuracy and the disentanglement of the latent representation of the sensing data. The training process involves optimizing the following objective function

$$\mathcal{L}(\theta, \phi; \mathbf{y}, \mathbf{z}, \beta) = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}\Big[\log p_\theta(\mathbf{y}|\mathbf{z})\Big] + \beta D_{KL}\Big(q_\phi(\mathbf{z}|\mathbf{y})||p(\mathbf{z})\Big). \tag{4.32}$$

The term $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}[\log p_\theta(\mathbf{y}|\mathbf{z})]$ in the objective function facilitates accurate reconstruction of the sensing data. Meanwhile, the Kullback-Leibler divergence, $D_{KL}(.||.)$, ensures that the latent space remains continuous and conforms to a standard multivariate Gaussian distribution. When $q_\phi(\mathbf{z}|\mathbf{y}) = p(\mathbf{z})$, the divergence D_{KL} is zero; otherwise, it increases monotonically as $q_\phi(\mathbf{z}|\mathbf{y})$ diverges from $p(\mathbf{z})$. Setting $\beta > 1$ enforces a stricter constraint on the latent space, promoting the learning of disentangled representations of the sensing data.

During training, the objective is to minimize the loss function in (4.32) by optimizing the

parameters ϕ and θ. That is,

$$\phi^*, \theta^* = \arg\min_{\phi,\theta} \mathcal{L}(\phi, \theta; \mathbf{Y}), \tag{4.33}$$

where $\mathbf{Y}$ denotes a $T \times N$ matrix representing a batch of T energy vectors. The Backpropagation (BP) algorithm [68] is utilized to efficiently compute the gradient of the loss in (4.32) with respect to ϕ and θ. However, the random sampling of $\mathbf{z}$, as illustrated in Fig. 4.3, renders direct backpropagation through the VAE nodes intractable. To address this, the reparameterization "trick" is employed, enabling a more stable and efficient training process [69]. This technique allows gradient flow by reformulating the sampling process as

$$\mathbf{z} = \mu_\phi + \sigma_\phi \odot \epsilon, \tag{4.34}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and $\odot$ is the element-wise product. The parameters of the VAE are then updated iteratively through the use of Stochastic Gradient Descent (SGD). During each iteration, a batch of energy vectors $\mathbf{Y}$ is utilized, and the SGD update is executed as follows

$$\phi^* := \phi - \eta \nabla_\phi \mathcal{L}(\phi, \theta; \mathbf{Y}), \tag{4.35}$$

$$\theta^* := \theta - \eta \nabla_\theta \mathcal{L}(\phi, \theta; \mathbf{Y}). \tag{4.36}$$

The step size for each iteration of SGD is controlled by the learning rate, denoted as η.

4.7.2 Affinity Propagation for Self-Organizing Clusters

The proposed VAE is designed to effectively learn a meaningful representation of the sensing data. However, in its standard form, it does not inherently facilitate direct clustering. To enable the FC to determine channel activity status (H_0/H_1), an unsupervised clustering algorithm is required. We utilize the Affinity Propagation (AP) algorithm, as it eliminates the need for cluster centroid initialization and prior knowledge of the number of clusters, enhancing the versatility and robustness of G-VAP. A detailed discussion of the AP algorithm [70] can be found in Section 4.6.1, with a summarized version provided in Algorithm 3 for reference.

Given a set of transformed energy vectors $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_L\}$ obtained through the encoder

network of the VAE, the AP algorithm identifies a set of exemplars (cluster representatives) and automatically determines the number of clusters in the data. AP operates based on a message-passing mechanism between data points. During training, the latent-space data points in $\mathbf{Z}$ iteratively exchange responsibility and availability messages until convergence. We adopt a convergence criterion that checks whether local decisions remain unchanged over a certain period. Additionally, we assign a uniform likelihood for all points to be selected as exemplars. At convergence, a data point is assigned to cluster H_0 if its similarity measure to H_0 is greater than its similarity to H_1. We use the negative Euclidean distance as the similarity measure.

Algorithm 3 Affinity Propagation algorithm.

1: Calculate the similarity $u(l,i) = -||l-i||$ for all pairs $\{l,i\} = 1,\ldots,L$ with $l \neq i$.
2: Initialize responsibility $r(l,i) = 0$ and availability $a(l,i) = 0$.
3: Set the preference value
4: **repeat**
5: Update responsibility:

$$r_t(l,i) \longleftarrow u(l,i) - \max_{i's.t.i' \neq i} \{a(l,i') + u(l,i')\}$$

6: Update availability:

$$a_t(l,i) \longleftarrow \min\left\{0, r(i,i) + \sum_{l's.t.l' \notin \{l,i\}} \max\{0, r(l',i)\}\right\}$$

7: For self-availability:

$$a_t(i,i) \leftarrow \sum_{l's.t.l' \neq i} \max(0, r(l',i))$$

8: Calculate $r_{t+1}(l,i)$ and $a_{t+1}(l,i)$ using (4.25) and (4.26)
9: Update iteration counter: $t \leftarrow t+1$
10: **until** convergence is reached
11: Assign exemplars based on the highest responsibility + availability values.

4.8 Simulation Results

In this section, we examine the performance of a CR network that leverages our proposed deep representation learning frameworks to boost CSS efficiency in both small- and large-scale cooperative networks.

4.8.1 Setup

We consider a cooperative CR network operating in an area of 1 km^2. The simulation parameters are summarized in Table. 4.1. The position coordinates of the n-th SU and the m-th PU are denoted by C_{SU}^{n} and C_{PU}^{m}, respectively. We consider the shadow fading $\psi_{m,n}$ component to be quasi-static throughout the sensing period. Initially, we set the path loss exponent to $\alpha = 4$ and the Nakagami-ν shape factor to $\nu = 1$. We then adjust these parameters to assess system performance across different propagation environments and fading conditions. We employ a sophisticated SGD method for updating parameters known as Adaptive Moment Estimation (Adam), which is favored for its accelerated computation speed. We compare our proposed unsupervised DL-based CSS approaches against supervised methods such as a fully connected Deep Neural Network (DNN) and a Long Short-Term Memory (LSTM) recurrent neural network. Table 4.2 provides an overview of the architectures of the proposed DL models, along with the benchmark models. However, it is important to mention that training supervised algorithms requires labeled data, which is impractical in real-world scenarios, as obtaining such labels would violate the fundamental principles of CR.
We consider the training loss of the supervised learning models to the binary cross entropy loss. The sensing data is split into training, validation, and testing sets containing 2400, 1000, and 10000 samples, respectively. To evaluate sensing performance, the Receiver Operating Characteristics (ROC) is used, which shows the probability of detection P_d as a function in the probability of false alarm P_{fa}. The ROC can be obtained by varying the decision threshold δ. Furthermore, Area Under

Table 4.1: Simulation parameters

Parameters		
Parameter	**Symbol**	**Value**
Number of PUs	m	[1:4]
Number of SUs	n	[1:9]
PU Transmit Power	ρ_m	200 mW
Bandwidth	ω	5 MHz
Sensing Period	τ	100 μs
Noise PSD	η	-174 dBm
β-VAE Parameter	β	1.5
Learning Rate	η	$[10^{-4} - 10^{-3}]$
Batch Size	T	100
Training Epoches	-	[100-120]

the ROC Curve (AUC) and testing accuracy were also chosen as performance evaluation metrics.

Table 4.2: Network architectures of the proposed deep learning models and baseline methods for CSS.

Network Type	Layers	Neurons in Each Layer	Hidden Activation	Output Activation
SAE (Proposed)	5	n,15,20,15,n	Tanh	None
β-VAE (Proposed)	6	n,3,2*2,2,3,n	Tanh	Sigmoid
DNN (Benchmark)	3	n,2,1	Tanh	Sigmoid
LSTM (Benchmark)	2	n, 3, 1	Sigmoid	Sigmoid

4.8.2 Results and Analysis

DeepSense

First, to illustrate the training process of the proposed SAE, Fig. 4.4 presents the training loss $\mathcal{L}_{\text{sparse}}$. The figure depicts both the training and validation loss per epoch, where an epoch represents a complete pass through the training set of energy vectors. Each epoch comprises multiple batches of sensing data. Observing the loss profile, we can infer that the training was effective, as both loss curves have converged. Furthermore, the minimal gap between the validation and training loss suggests that the model is well-fitted and exhibits strong generalization capabilities.

Fig. 4.5 evaluates the detection performance of the DeepSense approach across different latent

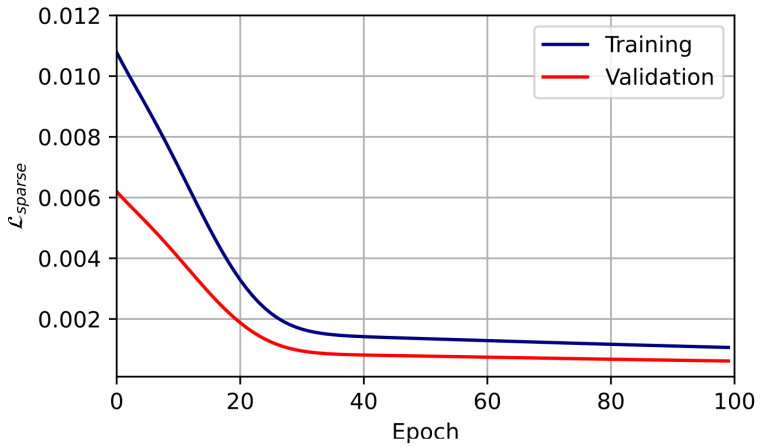


Figure 4.4: Training loss $\mathcal{L}_{\text{sparse}}$ of the proposed SAE.

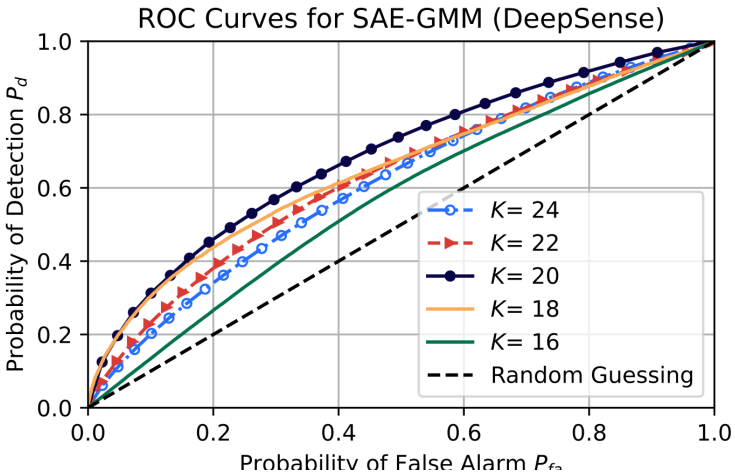


Figure 4.5: The effect of K on the detection performance of the DeepSense approach when n= 2, m= 2, C_1^{SU}= (1.5km,0.5km), C_2^{SU}= (-0.5km,0.5km), C_1^{PU}= (0km,0km), and C_2^{PU}= (0.5km,0.5km).

space dimensionalities K (i.e., the number of output neurons in the encoder). Analyzing the ROC curves, we observe that the highest detection performance is achieved at $K = 20$ neurons. This suggests that a 20-dimensional latent space is optimal for capturing the non-linear relationships among the reported energy levels, enabling the SAE to learn a meaningful sparse representation of the energy vectors. Consequently, the GMM can effectively cluster the sensing data and accurately determine the channel state (H_0/H_1).

Fig. 4.6 presents a performance comparison between the DeepSense detector ($K = 20$), the supervised DNN, and the unsupervised GMM. Since the DNN is trained in a supervised manner, the figure emphasizes the considerable performance difference between the DNN and the GMM. By employing an SAE to learn a representation of the sensing data in a higher-dimensional feature space, the DeepSense approach outperforms the GMM, which operates directly on the collected energy vectors. Moreover, the DeepSense approach achieves performance on par with the supervised DNN without requiring any labeled data, prior knowledge, or high SU cooperation costs. This makes our method both practical and suitable for the realistic deployment of learning-based CR systems.

Fig. 4.7 illustrates the effect of the number of intermittently active PUs (m) on the testing accuracy of the learning-based CR network. The testing accuracy is defined as the percentage of times the learning-based CR system correctly determines the channel state. As m increases, the accuracy of all learning methods improves. However, a notable performance gap is observed

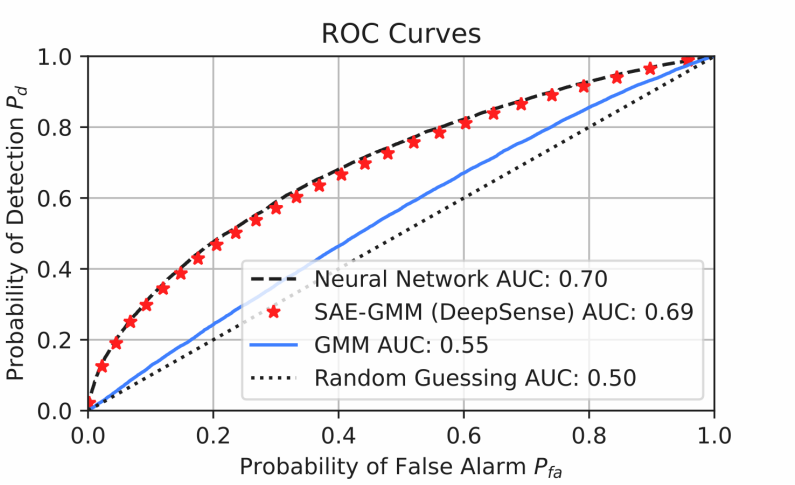


Figure 4.6: Benchmarking the detection performance of DeepSense when n= 2, m= 2, C_1^{SU}= (1.5km,0.5km), C_2^{SU}= (-0.5km,0.5km), C_1^{PU}= (0km,0km), and C_2^{PU}= (0.5km,0.5km).

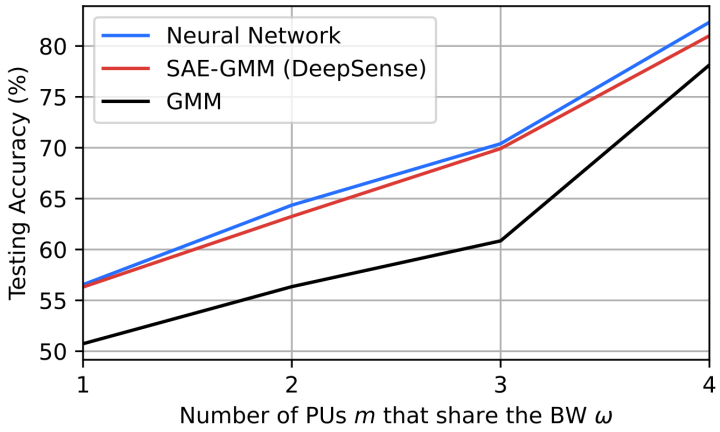


Figure 4.7: The effect of m on the detection performance of the intelligent radio network utilizing various learning approaches.

between the SAE-GMM and the GMM. Specifically, the CR network using the DeepSense detector demonstrates its ability to learn a valuable representation of the sensing data, which makes the detection performance more resilient to the number of intermittently active PUs.

DEAP Learning

Fig. 4.8 demonstrates the sensing performance of our proposed DEAP learning approach and compares it with our previously proposed DeepSense detector [10], the supervised DNN, and the unsupervised GMM. Unlike DeepSense, which relies on a GMM for clustering and requires prior

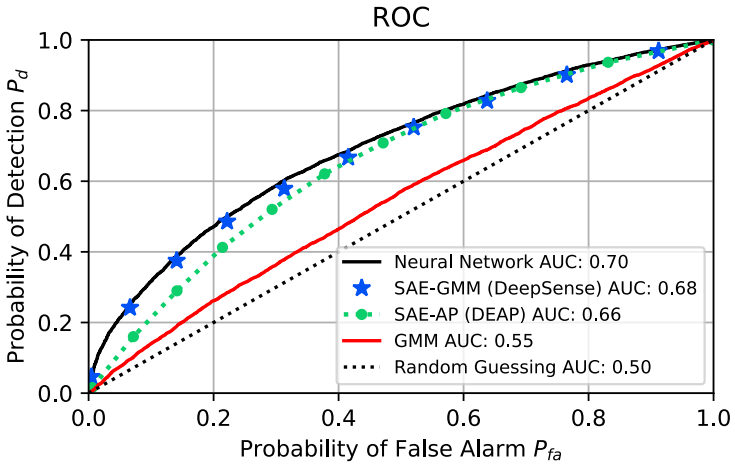


Figure 4.8: Benchmarking cooperative sensing performance of DEAP learning at $n = 2$, $m = 2$, C_1^{SU}= (1.5km,0.5km), C_2^{SU}= (-0.5km,0.5km), C_1^{PU} (0km,0km), and C_2^{PU}= (0.5km,0.5km).

knowledge such as the expected number of clusters and centroid initialization, DEAP learning achieves comparable performance without needing such prior information. The results in Fig. 4.8 show a significant performance gap between the DNN and the GMM, with the DEAP learning approach outperforming the GMM. Furthermore, DEAP learning attains performance similar to the supervised DNN without the use of labeled data, offering a practical and cost-effective solution for learning-based CR systems. These results underscore the effectiveness of the DEAP learning approach in improving the sensing performance of practical intelligent radio systems.

In Fig. 4.9, we examine the effect of increasing the number of cooperating SUs n on the performance of our DEAP learning approach for cooperative sensing. As n increases, there is a noticeable improvement in the AUC for all learning algorithms compared to Fig. 4.8. This improvement can be attributed to the higher number of features (energy levels) in the sensing data received by the FC, which enables it to benefit from the spatial diversity of the SUs, thereby providing the system with more degrees of freedom. With this enhancement, our DEAP learning approach continues to outperform the GMM, achieving performance comparable to both DeepSense and the supervised DNN. These results highlight the effectiveness of our approach in exploiting the advantages of cooperation among SUs, while maintaining competitive performance with state-of-the-art techniques.

Fig. 4.10 assesses the detection performance of the proposed DEAP learning approach at different

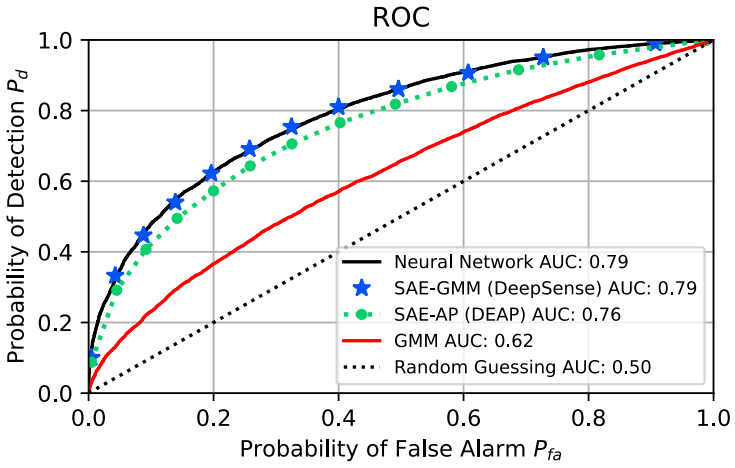


Figure 4.9: Benchmarking cooperative sensing performance of DEAP learning at $n = 4$, $m = 2$, C_1^{SU}= (1.5km,0.5km), C_2^{SU}= (-0.5km,0.5km), C_3^{SU}= (0km,1km), C_4^{SU}= (1km,0km), C_1^{PU} (0km,0km), and C_2^{PU}= (0.5km,0.5km).

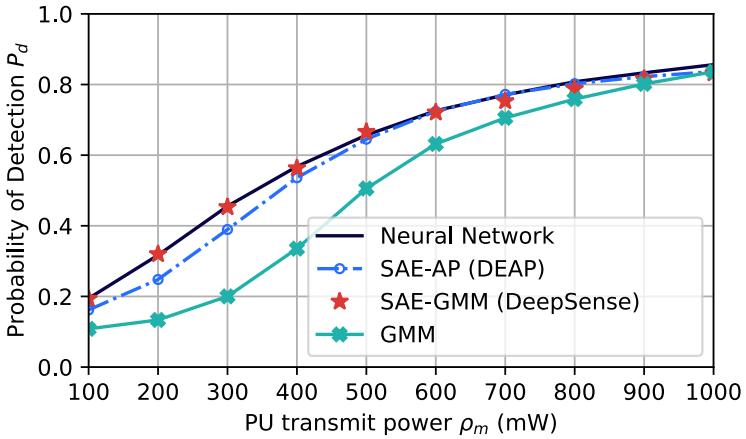


Figure 4.10: Effect of ρ_m on detection probability at P_{fa}= 0.1, $n = 2$, $m = 2$, C_1^{SU}= (1.5km,0.5km), C_2^{SU}= (-0.5km,0.5km), C_1^{PU} (0km,0km), and C_2^{PU}= (0.5km,0.5km).

PU transmission power levels ρ_m. As shown in Fig. 4.10, the probability of detection P_d increases as the ρ_m values rise. With higher ρ_m, the energy levels of the spectrum also increase, making it easier for the CR network to distinguish between an unoccupied and an occupied channel. Importantly, the P_d of the proposed DEAP learning approach closely aligns with that of the supervised DNN, even though it does not rely on labeled data. At low ρ_m, all learning techniques exhibit similar performance. However, as ρ_m increases, the DEAP approach outperforms the GMM, achieving performance close to the supervised DNN.

Fig. 4.11 shows the detection performance of the CR network as the number of intermittently active PUs, denoted by m, increases from 2 to 3. Notably, the proposed DEAP learning approach achieves performance comparable to that of DeepSense and the supervised DNN. Furthermore, as m increases, there is a corresponding rise in the AUC across all learning algorithms, compared to the results in Fig. 4.8. This improvement is due to the higher spectrum energy levels detected, which lead to more distinguishable clusters (H_0/H_1) in the representation space. These findings demonstrate that the CR network employing DEAP learning can effectively learn a valuable representation of the sensing data, ensuring robust detection performance despite variations in the number of intermittently active PUs.

To assess the performance of our proposed DEAP learning approach in different propagation environments, we vary the path loss exponent α. Specifically, we simulate scenarios with $\alpha = 2.42$ for outdoor line-of-sight (OL) environments, $3.5 < \alpha \leq 4$ for non-line-of-sight (NLOS) environments, and $\alpha = 4.5$ for obstructed environments, such as those with buildings. As shown in Fig. 4.12, the detection performance decreases as α increases, reflecting the increasing difficulty of the environment, with the poorest performance observed in obstructed environments. However, the DEAP learning approach maintains strong performance in both OL and NLOS environments.

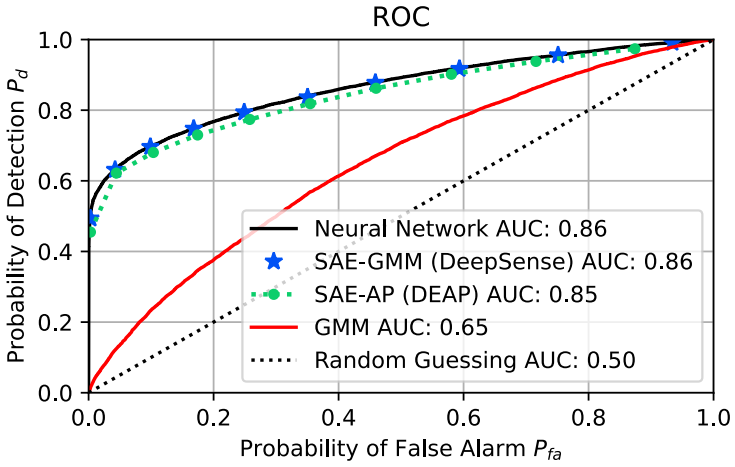


Figure 4.11: Cooperative sensing performance at $n = 2$, $m = 3$, C_1^{SU}= (1.5km,0.5km), C_2^{SU}= (-0.5km,0.5km), C_1^{PU} (0km,0km), C_2^{PU}= (0.5km,0.5km), and C_3^{PU}= (-0.5km,0km).

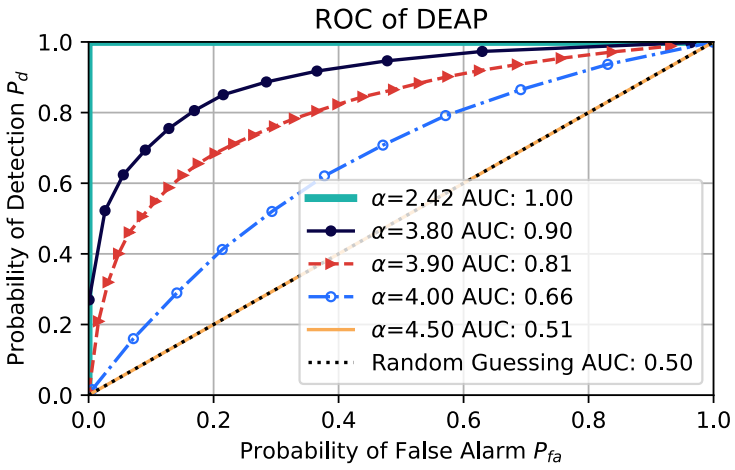


Figure 4.12: Effect of propagation environment on cooperative sensing performance of DEAP learning at $n = 2$, $m = 2$, C_1^{SU}= (1.5km,0.5km), C_2^{SU}= (-0.5km,0.5km), C_1^{PU} (0km,0km), and C_2^{PU}= (0.5km,0.5km).

G-VAP

Fig. 4.13 evaluates the detection performance of the learning-based CR network using our G-VAP method across different latent space dimensions K. From the analysis of the ROC curves, it is clear that selecting $K = 2$ yields the best detection performance. This is because increasing the dimensionality of the latent space beyond this value may not sufficiently constrain the β-VAE model during training, potentially leading to overfitting and reducing its effectiveness. Therefore, a latent space with 2 dimensions is found to be the most effective for capturing the complex relationships among the reported energy levels, enabling the proposed β-VAE to learn a meaningful representation of the energy vectors in a lower-dimensional latent space. Consequently, the AP algorithm at the FC can accurately cluster the sensing data, efficiently determining the channel state (H_0/H_1).

Fig. 4.14 compares the detection performance of the unsupervised G-VAP detector ($K = 2$) with the supervised DNN and LSTM, along with the vanilla AP algorithm trained on high-dimensional sensing data, against our proposed G-VAP approach. As shown, there is a significant performance gap between the supervised DNN and LSTM and the AP algorithm. The unsupervised AP algorithm, when trained on high-dimensional data, faces challenges in creating an effective clustering model. In contrast, the G-VAP method, utilizing a β-VAE to learn a representation of the sensing data in a lower-dimensional latent space, outperforms the vanilla AP algorithm. Furthermore, the proposed G-VAP approach achieves performance comparable to the supervised DNN and LSTM, without requiring

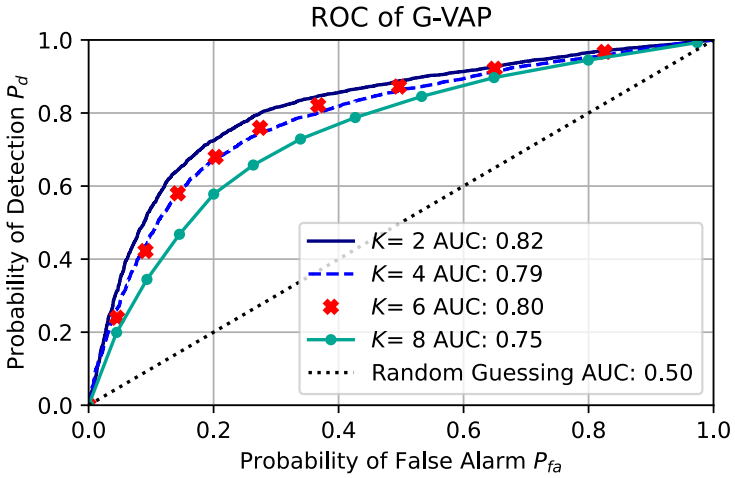


Figure 4.13: The effect of latent dimensionality K on the performance of G-VAP at n= 9, m= 1, and C_1^{PU}= (0.5km,0.5km).

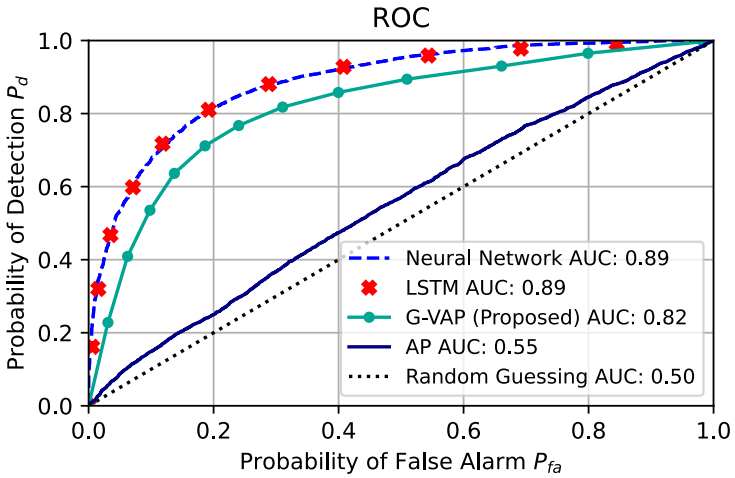


Figure 4.14: Benchmarking G-VAP against other learning strategies at n= 9, m= 1, and C_1^{PU}= (0.5km,0.5km).

labeled data, prior knowledge, or the increased costs of SU-PU cooperation. This efficiency makes our method a promising and effective solution for real-world applications of unsupervised deep learning in CR systems.

Fig. 4.15 illustrates the sensing performance of the G-VAP approach across various PU transmission power levels, ρ_m. The figure reveals a positive correlation between the detection probability P_d and the increase in transmission power ρ_m. As ρ_m increases, the spectral energy rises, enabling the FC to more effectively distinguish between an empty channel with only noise and an occupied

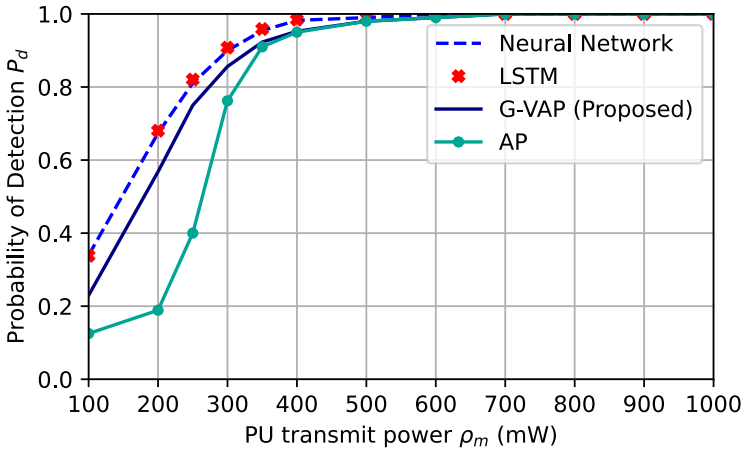


Figure 4.15: The effect of ρ_m on the detection performance when P_{fa}= 0.1.

channel. Notably, the detection performance of the G-VAP method closely matches that of the supervised DNN and LSTM, irrespective of the value of ρ_m, unlike the vanilla AP. At higher ρ_m levels, the performance of all learning models converges, as the clusters H_0/H_1 become more separable.

Fig. 4.16 shows how the testing accuracy of the learning-based CR network is affected by the number of intermittently active PUs, denoted as m. Testing accuracy is defined as the percentage of times the learning-based CR system correctly determines the channel state. As m increases, a noticeable improvement in accuracy is observed across all learning methods. This is due to the increase in spectrum energy levels as more PUs use the channel, which enhances the separability of clusters H_0 and H_1. However, the G-VAP approach exhibits a significant advantage over the vanilla AP algorithm and achieves performance comparable to the supervised DNN and LSTM, without requiring labeled data for training. The CR network employing G-VAP is highly effective in learning a useful representation of the sensing data, ensuring robust detection performance despite the varying number of intermittently active PUs.

To evaluate the performance of the G-VAP approach under different propagation conditions, we adjust the path loss exponent α. We use $\alpha = 2.42$ to represent outdoor line-of-sight (OL) settings, $3.5 < \alpha \leq 4$ for non-line-of-sight (NLOS) environments, and $\alpha = 4.5$ for obstructed conditions such as buildings. As shown in Fig. 4.17, an increase in α results in decreased detection performance, with the most challenging scenarios arising in obstructed environments. Despite this, the G-VAP

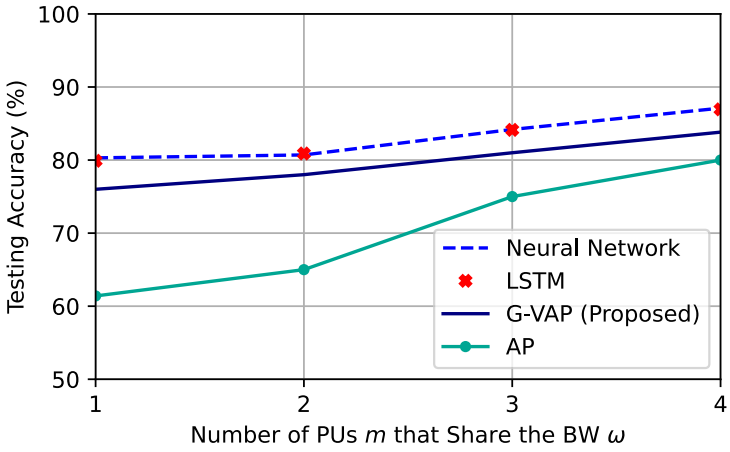


Figure 4.16: The effect of the number of PUs m on G-VAP's performance.

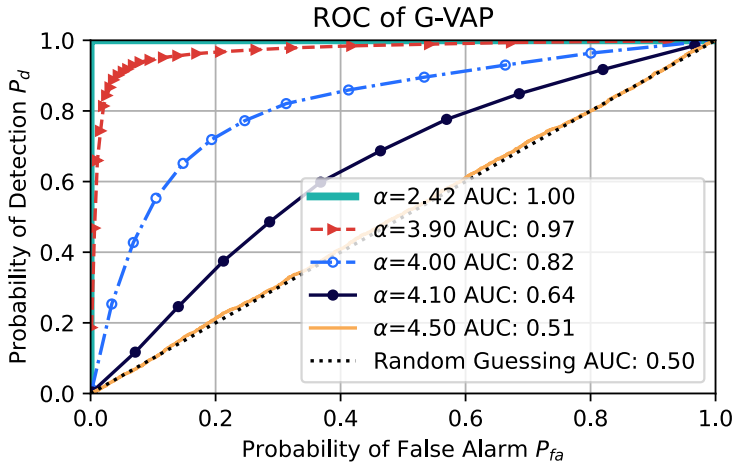


Figure 4.17: The effect of path loss exponent α on the performance of G-VAP.

approach continues to deliver strong performance in both OL and NLOS settings.

To further assess the G-VAP approach in fading conditions, we fix $\alpha = 4$ and vary the fading severity ν. Fig. 4.18 illustrates that higher ν values reduce the fading effects within the channel, improving sensing performance. Conversely, lower ν values lead to stronger fading, causing a decline in performance. Despite the fading challenges, the G-VAP approach remains resilient, maintaining reliable sensing capabilities. This is attributed to the β-VAE's ability to perform both preprocessing and feature learning, which enables the FC to effectively mitigate the negative impact of fading.

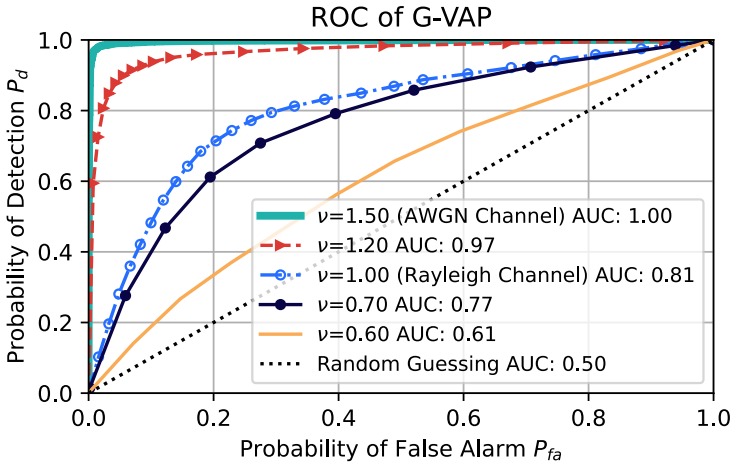


Figure 4.18: The effect of Nakagami-ν shape factor on the performance of G-VAP.

4.9 Conclusions

In this chapter, we introduced three deep representation learning frameworks—DeepSense, DEAP learning, and G-VAP—designed to enhance spectrum reasoning and analysis for CSS in CR networks. These frameworks leverage unsupervised DL techniques to effectively learn representations of sensing data, addressing key challenges associated with both limited and large-scale SU cooperation. Moreover, they address key challenges in representation learning and clustering without requiring labeled data or prior knowledge of signal distributions. DeepSense employs an SAE to learn representations of sensing data, combined with GMM clustering. Building upon DeepSense, DEAP learning was introduced to overcome the limitations associated with GMM-based clustering, particularly its sensitivity to cluster centroid initialization and the need for predefined cluster counts. DEAP learning retains the SAE for representation learning but employs the AP algorithm for clustering. Unlike traditional clustering methods, AP infers the number of clusters directly from the data and does not require predefined centroids, making DEAP learning more adaptable to dynamic spectrum environments. To address challenges associated with a larger pool of cooperating users and the resulting high-dimensional energy vectors, we introduced G-VAP, the first fully unsupervised deep generative framework for CSS. G-VAP utilizes a β-VAE to learn disentangled latent representations of high-dimensional sensing data, allowing for more precise and separable

clustering. Unlike previous DL-based approaches for CSS, the proposed methods require significantly less training data, thereby reducing communication overhead between the SU and the FC. Furthermore, extensive simulations across diverse network settings, propagation environments, and fading conditions demonstrate that the proposed approaches are on par with supervised DL-based methods and outperform non-DL techniques, highlighting their effectiveness.

Chapter 5

Distributed Learning for Large-Scale Mobile Spectrum-Aware Networks

5.1 Introduction

With the emergence of new wireless networks and applications, CR networks are expanding to encompass larger regions, often involving multiple PUs. Due to factors such as path loss, shadowing, and fading, the spectrum state perceived by users varies across different locations within the network, depending on whether the SUs are within or beyond the transmission range of the PUs. This variation complicates accurate spectrum sensing, particularly in non-cooperative scenarios. To enhance spectrum state identification, Cooperative Spectrum Sensing (CSS) was introduced, enabling spatially distributed users to collaborate and share sensing information to improve detection accuracy. However, these approaches generally assume that users remain stationary and require a significant number of SUs. Additionally, aside from the extensive cooperation and data exchange required among SUs, transmitting sensing data may unintentionally reveal private user information, posing security and privacy risks.

5.2 Related Works

With advancements in neural networks, Deep Learning (DL) has demonstrated significant advantages in feature extraction, leading to notable improvements in spectrum sensing [18]. A semi-supervised spectrum sensing approach utilizing a Variational Autoencoder (VAE) trained with a Gaussian mixture prior was introduced in [53]. Likewise, [71] proposed a semi-supervised spectrum sensing method that integrates a Generative Adversarial Network (GAN) with a Convolutional Neural Network (CNN) to enhance robustness at low Signal-to-Noise Ratios (SNRs). In [11], we developed an unsupervised CSS technique that employs a Sparse Autoencoder (SAE) for sensing data representation learning, followed by clustering using Affinity Propagation (AP). The work in [72] presented a graph neural network-based CSS method, which transforms signals into graph topology to capture latent structural relationships, thereby improving performance under noise uncertainty. Additionally, [73] introduced a CNN-based dequantization method that enhances CSS by converting low-bit sensing data into near full-precision values without incurring signaling overhead. A Federated Learning (FL)-based CSS framework was proposed in [43], enabling SUs to train a supervised deep neural network locally and transmit gradients to a Fusion Center (FC) instead of raw data. Furthermore, [74] presented a compressed sensing-based FL framework designed to improve data aggregation efficiency while preserving privacy.

The growing integration of DL-based CR operations, as demonstrated in the aforementioned studies, underscores their essential role in facilitating network autonomy. However, DL techniques for CSS, such as those in [43, 53], depend on labeled training data, which is difficult to obtain in CR networks where users perform blind sensing without prior channel knowledge and lack direct communication with PUs. Furthermore, existing DL-based CSS approaches typically assume that all SUs remain stationary. In large-scale networks, this assumption necessitates deploying a substantial number of fixed SUs for spectrum sensing, which is often impractical or cost-prohibitive. Notably, mobility is an intrinsic feature of wireless network users, and prior research has demonstrated that user mobility can significantly enhance spatial-temporal diversity, thereby improving received signal quality across different wireless environments. Additionally, studies such as [10, 11, 73, 75] employ a centralized learning-based CSS framework, requiring SUs to transmit their sensing data to a central

entity for spectrum state identification. However, this centralized approach raises privacy concerns and imposes substantial communication overhead.

5.3 Contributions

Motivated by the above, our contributions can be summarized as follows:

- We propose FeRAP, the first fully unsupervised deep FL approach for robust, distributed, and secure CSS in large-scale mobile networks. By leveraging the mobility of a few SUs across a wide geographical area, spectrum data is gathered locally and used to train a joint model. Each SU then transmits its model parameters rather than spectrum data to the FC. This distributed approach not only reduces communication overhead between the FC and SUs but also enhances model performance by leveraging the users' spatial diversity.

- A novel β-Variational Autoencoder (β-VAE) architecture is proposed that identifies independent latent variables and learns disentangled representations of the sensing data in a lower-dimensional space. In the training process, only a small amount of unlabeled raw spectrum data is required.

- Affinity Propagation (AP) is then locally trained on the learned representations at each co-operating user, allowing the SUs to infer the spectrum state automatically without requiring prior knowledge of the number of spectrum states or cluster centroid initializations.

- We conduct extensive experiments that validate the effectiveness and scalability of FeRAP, highlighting its superior performance over other deep learning and FL CSS methods.

5.4 System Model

We examine the large-scale CR network depicted in Fig. 5.1, which comprises $n = 1, ..., N$ mobile SUs and $m = 1, ..., M$ PUs, each constrained by a limited transmission range. The number of active PUs is unknown in advance. The spectrum consists of B channels ($B \geq M$), each with a bandwidth of ω, where each active PU intermittently operates on one of these channels. Detecting

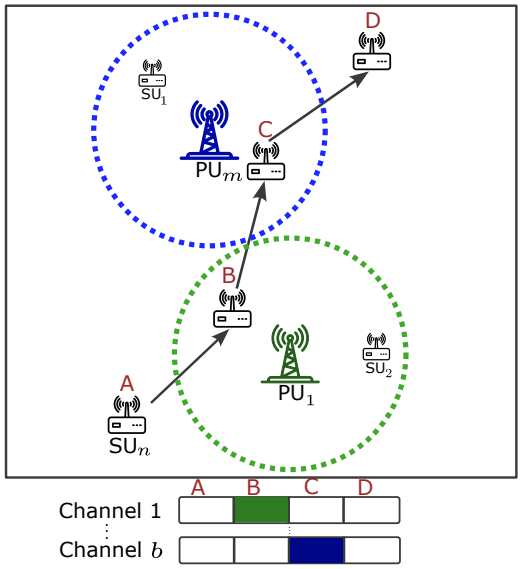


Figure 5.1: The studied large-scale mobile CR network. Dashed lines represent the transmission range of each PU. At various sensing points along a path (A,B,C,D), the n-th mobile SU encounters different occupancy states of the primary network. Colored channels indicate those occupied by the respective PU.

PU activity is formulated as a binary hypothesis problem, where H_0 represents an idle channel, and H_1 indicates an occupied channel. If channel b is in use, the channel occupancy indicator s_b is set to 1 (H_1); otherwise, it remains 0 (H_0). The N SUs move at low speeds within the network, allowing the Doppler effect to be neglected [75]. Typically, SUs follow predefined routes that account for the geographical structure of the network, with each route being segmented into $l = 1, ..., L$ sensing locations.

As each SU moves, it gathers spectrum data by measuring the energy level of channel b using a basic energy detector, storing the results locally. Each SU remains at a sensing location l for τ seconds, during which it accumulates $\omega\tau$ energy for channel b. Throughout the sensing period, we assume that $h_{m,n}$ remains constant, a reasonable assumption given that the sensing duration can be designed to be shorter than the channel coherence time. The energy sample recorded at the i-th instance by the n-th SU for channel b is

$$E_{n,b}(i) = s_b h_{m,n} X_m(i) + N_n(i), \tag{5.1}$$

where $X_m(i)$ is the transmitted m-th PU signal. The channel gain between the m-th PU and n-th SU, $h_{m,n}$, is modeled using a Nakagami-ν distribution, suitable for outdoor multipath fading. Its

Probability Density Function (PDF) is given by

$$f_{h_{m,n}}(y;\nu,\Omega) = \frac{2}{\Gamma(\nu)}(\frac{\nu}{\Omega})^{\nu} y^{2\nu-1} \exp\left(-\frac{\nu y^2}{\Omega}\right), \tag{5.2}$$

where $\nu \geq 0.5$ denotes the Nakagami shape parameter, which quantifies the severity of fading. The spread parameter is given by $\Omega = E[h_{m,n}^2] > 0$, while $\Gamma(.)$ represents the Gamma function. The thermal noise $N_n(i)$ at the n-th SU follows a Gaussian distribution $\mathcal{N}(0, \sigma_n^2)$, with a power spectral density expressed as $\sigma_n^2 = E[|N_n(i)|^2]$. As a result, the energy level of channel b at the n-th SU, normalized by σ_n^2, is

$$y_{n,b} = \frac{2}{\sigma_n^2} \sum_{i=1}^{\omega\tau} |E_{n,b}(i)|^2. \tag{5.3}$$

$y_{n,b}$ follows a non-central chi-squared distribution with $q = 2\omega\tau$ degrees of freedom and a non-centrality parameter $\zeta_{n,b}$ defined as

$$\zeta_{n,b} = \frac{2\tau}{\sigma_n^2} s_b g_{m,n} \rho_m, \tag{5.4}$$

where ρ_m represents the transmit power of the m-th PU, expressed as

$$\rho_m = \frac{\sum_{i=1}^{\omega\tau} E[|X_m(i)|^2]}{\tau}. \tag{5.5}$$

$g_{m,n}$ represents the power attenuation from the m-th PU to the n-th SU defined as

$$g_{m,n} = |h_{m,n}|^2 = D_{m,n}^{-\alpha} \cdot \psi_{m,n} \cdot \nu_{m,n}, \tag{5.6}$$

where $D_{m,n}$ is the Euclidean distance, α is the path loss exponent, $\psi_{m,n}$ accounts for shadow fading, and $\nu_{m,n}$ represents the multipath fading component.

Normally, if the number of energy samples $\omega\tau$ is large, the energy level $y_{n,b}$ observed by the n-th

SU in channel b can be modelled as a Gaussian distribution $\mathcal{N}(\mu_{y_{n,b}|s_b}, \sigma^2_{y_{n,b}|s_b})$, with parameters

$$\mu_{y_{n,b}|s_b} = \mathbb{E}[y_{n,b}|s_b] = 2\omega\tau + \frac{2\tau}{\sigma_n^2} s_b g_{m,n} \rho_m, \tag{5.7}$$

$$\sigma^2_{y_{n,b}|s_b} = \mathbb{E}[(y_{n,b} - \mu_{y_{n,b}|s_b})^2|s_b] = 4\omega\tau + \frac{8\tau}{\sigma_n^2} s_b g_{m,n} \rho_m. \tag{5.8}$$

As the n-th SU moves along its path while sensing channel b, the energy level recorded at the l-th location is denoted as $y^l_{n,b}$. Given a total of L sensing locations along the path, the complete spectrum sensing data gathered by the n-th SU is represented as an energy vector $\mathbf{y}_{n,b} = \{y^1_{n,b}, ..., y^L_{n,b}\}$. It is important to note that SUs do not require synchronization during the sampling process, and the number of sensing locations may vary among them. However, for simplicity, we assume that all SUs have the same number of sensing locations.

5.5 FeRAP: A Deep Federated Representation Learning Approach for Secure and Distributed Cooperative Sensing

Federated Learning (FL) is a decentralized machine learning framework that leverages local sensing data from each SU to collaboratively train a model, enhancing overall performance. In this study, we propose FeRAP, a novel deep federated representation learning approach for mobile CSS, as illustrated in Fig. 5.2(a). Due to the challenge of obtaining labeled data in CR networks, FeRAP operates in an unsupervised fashion, relying solely on unlabeled sensing data for training. It utilizes an advanced variant of generative Variational Autoencoders (VAEs) to model the sensing data distribution while simultaneously performing non-linear compression of the high-dimensional input space. This not only enhances sensing accuracy but also facilitates the generation of previously unseen synthetic data. During each training iteration, SUs transmit their model parameters to the Fusion Center (FC), which aggregates them and distributes the updated parameters back. This collaborative training strategy preserves the privacy of SUs while enabling them to benefit from shared insights without directly exposing their raw data.

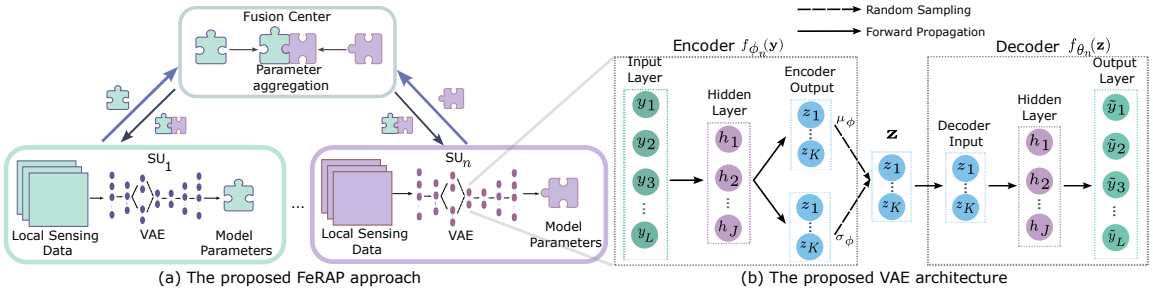


Figure 5.2: The proposed FeRAP approach and the VAE architecture at the n-th SU.

5.5.1 Deep Federated Representations through β-VAE

A Variational Autoencoder (VAE) is a deep probabilistic generative model that employs variational Bayesian inference to learn a probabilistic representation of sensing data in a latent space. We propose a VAE architecture built using Deep Neural Networks (DNNs), as depicted in Fig. 5.2(b), consisting of a probabilistic encoder (inference model) and a probabilistic decoder (generative model). The decoder is trained locally at each SU n to model the joint distribution $p_{\theta_n}(\mathbf{y}_{n,b}, \mathbf{z}_{n,b}) = p_{\theta_n}(\mathbf{y}_{n,b}|\mathbf{z}_{n,b})p_{\theta_n}(\mathbf{z}_{n,b})$, where $p_{\theta_n}(\mathbf{z}_{n,b}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a prior. In VAEs, computing the exact posterior distribution $p_{\theta_n}(\mathbf{z}_{n,b}|\mathbf{y}_{n,b})$ is often infeasible due to its complexity [53]. To address this, variational inference approximates the posterior using $q_{\phi_n}(\mathbf{z}_{n,b}|\mathbf{y}_{n,b})$, where the probabilistic encoder estimates the latent variables from the observed data $\mathbf{y}_{n,b}$. The prior $p_{\theta_n}(\mathbf{z}_{n,b})$ plays a crucial role in encouraging disentanglement within $q_{\phi_n}(\mathbf{z}_{n,b}|\mathbf{y}_{n,b})$. Ensuring alignment with the prior effectively manages the latent capacity and promotes statistical independence. During training, both ϕ_n and θ_n are optimized to minimize divergence between $q_{\phi_n}(\mathbf{z}_{n,b}|\mathbf{y}_{n,b})$ and $p_{\theta_n}(\mathbf{z}_{n,b}|\mathbf{y}_{n,b})$. The encoder produces the parameters μ_{ϕ_n} and σ_{ϕ_n} for the distribution $q_{\phi_n}(\mathbf{z}_{n,b}|\mathbf{y}_{n,b})$, which is defined as

$$\mathbf{z}_{n,b} \sim q_{\phi_n}(\mathbf{z}_{n,b}|\mathbf{y}_{n,b}) = \mathcal{N}(\mathbf{z}_{n,b}|\mu_{\phi_n}, \sigma^2_{\phi_n}\mathbf{I}). \tag{5.9}$$

The proposed VAE architecture at each SU n consists of a fully connected DNN encoder. This encoder is parameterized by $\phi_n = \{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}_{i=1,2}$, where $\mathbf{W}^{(i)}$ and $\mathbf{b}^{(i)}$ denote the weight matrices and biases, and the superscripts i refer to the layer numbers. The input layer of the encoder contains L neurons, corresponding to the energy levels gathered along the SU's path. The encoder has a hidden layer with J neurons, where $J < L$. The output of the encoder consists of two

components, μ_{ϕ_n} and σ_{ϕ_n}, each represented by K neurons, where $K < J < L$. The encoder's output can therefore be expressed as

$$f_{\phi_n}(\mathbf{y}_{n,b}) = \mu_{\phi_n}, \sigma_{\phi_n} = \Big(\tanh(\mathbf{y}_{n,b}\mathbf{W}^{(1)} + \mathbf{b}^{(1)}) \Big)\mathbf{W}^{(2)} + \mathbf{b}^{(2)}. \tag{5.10}$$

In each VAE, the hyperbolic tangent activation function, $\tanh(.)$, is used to capture the non-linear relationships in the energy levels collected by an SU. The choice of $\tanh$ is based on its symmetry around zero, which facilitates faster convergence. The decoder's architecture consists of a DNN with three layers containing K, J, and L neurons, respectively. The decoder's parameters are $\theta_n = \{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}_{i=3,4}$, and its purpose is to reverse the encoder's function by reconstructing the energy vector $\tilde{\mathbf{y}}_{n,b}$ from the input $\mathbf{z}_{n,b}$. Therefore, the output $\tilde{\mathbf{y}}_{n,b}$ is defined as

$$f_{\theta_n}(\mathbf{z}_{n,b}) = \tilde{\mathbf{y}}_{n,b} = \sigma_{\text{sigmoid}}\Big(\tanh(\mathbf{z}_{n,b}\mathbf{W}^{(3)} + \mathbf{b}^{(3)}) \Big)\mathbf{W}^{(4)} + \mathbf{b}^{(4)}, \tag{5.11}$$

where $\sigma_{\text{sigmoid}}(x)$ is the sigmoid function, defined as $\sigma_{\text{sigmoid}}(x) = \frac{1}{1+e^{-x}}$, ensuring that the VAE outputs are constrained between 0 and 1.

We utilize an advanced variant of VAEs, known as β-VAE, which incorporates a hyperparameter β to regulate the trade-off between reconstruction accuracy and the disentanglement of the latent representation of the sensing data. The training objective at each SU is therefore

$$\mathcal{L}(\theta_n, \phi_n; \mathbf{y}_{n,b}, \mathbf{z}_{n,b}, \beta) = -\mathbb{E}_{q_{\phi_n}(\mathbf{z}_{n,b}|\mathbf{y}_{n,b})}[\log p_{\theta_n}(\mathbf{y}_{n,b}|\mathbf{z}_{n,b})] + \beta D_{KL}(q_{\phi_n}(\mathbf{z}_{n,b}|\mathbf{y}_{n,b})||p_{\theta_n}(\mathbf{z}_{n,b})). \tag{5.12}$$

The term $\mathbb{E}_{q_{\phi_n}(\mathbf{z}_{n,b}|\mathbf{y}_{n,b})}[\log p_{\theta_n}(\mathbf{y}_{n,b}|\mathbf{z}_{n,b})]$ encourages the accurate reconstruction of the sensing data, while $D_{KL}(.||.)$ denotes the Kullback-Leibler divergence, which ensures that the latent space remains continuous and conforms to a standard multivariate Gaussian distribution. If $q_{\phi_n}(\mathbf{z}_{n,b}|\mathbf{y}_{n,b}) = p_{\theta_n}(\mathbf{z}_{n,b})$, then $D_{KL} = 0$; otherwise, D_{KL} increases as $q_{\phi_n}(\mathbf{z}_{n,b}|\mathbf{y}_{n,b})$ deviates from $p_{\theta_n}(\mathbf{z}_{n,b})$. By setting $\beta > 1$, a stronger constraint is applied to the latent space, promoting disentangled representations of the sensing data. The training objective of each SU is to minimize

the loss function in (5.12) by optimizing ϕ_n and θ_n.

$$\phi_n^*, \theta_n^* = \arg\min_{\phi_n, \theta_n} \mathcal{L}(\phi_n, \theta_n; \mathbf{Y}_{n,b}). \tag{5.13}$$

Here, $\mathbf{Y}_{n,b}$ represents a $T \times L$ matrix, containing a batch of T energy vectors. The Backpropagation (BP) algorithm [68] is employed to compute the gradient of the loss in (5.12) with respect to ϕ_n and θ_n efficiently. However, the random sampling of $\mathbf{z}_{n,b}$, as shown in Fig. 5.2(b), makes direct Backpropagation (BP) through the VAE impractical. To address this, the reparameterization "trick" is utilized, which facilitates a more stable and efficient training process [69], by reformulating the sampling procedure as

$$\mathbf{z}_{n,b} = \mu_{\phi_n} + \sigma_{\phi_n} \odot \epsilon, \tag{5.14}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and $\odot$ denotes the element-wise product. Parallel gradient descent is used to iteratively update the parameters of the VAEs at each SU. In each iteration, an SU processes a batch of energy vectors $\mathbf{Y}_{n,b}$, and as a result, the parallel Stochastic Gradient Descent (SGD) update for all N SUs is performed as

$$\phi_n^* := \phi_n - \eta \nabla_{\phi_n} \mathcal{L}(\phi_n, \theta_n; \mathbf{Y}_{n,b}), \tag{5.15}$$

$$\theta_n^* := \theta_n - \eta \nabla_{\theta_n} \mathcal{L}(\phi_n, \theta_n; \mathbf{Y}_{n,b}). \tag{5.16}$$

The step size for each SGD iteration is determined by the learning rate η. In this work, we use a refined SGD method called Adaptive Moment Estimation (Adam), preferred for its faster computation speed [9, 53].

At each iteration, the SUs update their VAE parameters ϕ_n and θ_n and send them to the FC for aggregation. To improve security, differential privacy and secret sharing techniques can be applied to obscure any parameters sent [43]. The FC aggregates the parameters as $\phi_n = \frac{\sum_{n=1}^{N} \phi_n}{N}$ and $\theta_n = \frac{\sum_{n=1}^{N} \theta_n}{N}$, then sends them back to the SUs for model refinement as shown in Fig. 5.2(a). Unlike traditional DL-based CSS, FeRAP transmits model parameters rather than data, reducing communication overhead and ensuring the privacy of each SU while maintaining control over the training process. After training is complete, all SUs share the final model parameters.

It is noteworthy that SUs only require the encoder $f_{\phi_n}(.)$ to map newly collected energy vectors onto the latent space. The decoder $f_{\theta_n}(.)$, on the other hand, is solely used to generate synthetic samples based on the learned data distribution within the latent space. If this functionality is not needed, SUs may discard the decoder's parameters without affecting their ability to process new data. An additional advantage of our proposed approach is that a newly joining SU can simply download the β-VAE parameters from the FC and immediately use them to transform its collected spectrum data. It is also important to highlight that learning algorithms typically assume that the data collected at each SU is independent and identically distributed (i.i.d.). However, in real-world scenarios, this assumption may not hold, potentially causing the learning model to struggle with generalization once deployed. Furthermore, it is generally assumed that the data distribution remains stable after deployment. If the distribution changes, leading to a situation known as dataset shift due to a non-stationary environment, it may become necessary to retrain the model to maintain performance.

5.5.2 Clustering with Affinity Propagation via Message Passing

The proposed VAE is structured to effectively capture a representation of the sensing data in latent space at each SU. However, it is essential to note that the VAE does not inherently support clustering. Therefore, an unsupervised clustering algorithm is required for each SU to identify the spectrum state (H_0/H_1). The dynamic nature of the radio environment makes unsupervised clustering challenging, especially for widely used methods like k-centers algorithms. These approaches begin with randomly selected exemplars and iteratively refine them to minimize the sum of squared errors. However, their high sensitivity to the initial choice of exemplars often necessitates multiple runs with different starting points to achieve reliable results. Additionally, these methods require prior knowledge of the expected number of clusters/spectrum states.

To overcome these limitations, we use the Affinity Propagation (AP) algorithm, as detailed in Section 4.6.1, for clustering. The AP algorithm seeks to identify a representative set of data points, called exemplars, within the encoder's latent space and assigns labels to the remaining data points based on their proximity to these exemplars. By representing each data point as a node in a network, the AP algorithm facilitates the exchange of real-valued messages (responsibility and availability)

between nodes until an optimal set of exemplars and clusters is determined. There are two types of messages exchanged between data points, each considering a different aspect of competition. The "responsibility" $r(l, i)$, illustrated in Fig. 5.3(a), is sent from data point l to candidate exemplar point i, indicating the accumulated evidence of how well-suited point i is as the exemplar for point l, considering other potential exemplars. The "availability" $a(l, i)$, shown in Fig. 5.3(b), is sent from candidate exemplar point i to point l, reflecting the accumulated evidence of how appropriate it is for point l to select point i as its exemplar, while accounting for the support from other points suggesting that point i should be an exemplar. The AP algorithm is presented in detail in Algorithm 3.

Let $\mathbf{Z}_{n,b} = \{\mathbf{z}^1_{n,b}, \ldots, \mathbf{z}^T_{n,b}\}$ be the set of T transformed energy vectors by the n-th SU's encoder $f_{\phi_n}(\mathbf{Y}_{n,b})$. Let $\mathbf{z}^0_{n,b}$ and $\mathbf{z}^1_{n,b}$ denote the exemplars for clusters H_0 and H_1, respectively. We use the negative Euclidean distance as a similarity metric, where the similarity between points in $\mathbf{Z}n, b$ is defined as $u(l, i) = -||\mathbf{z}^l n, b - \mathbf{z}^i_{n,b}||$ for $l, i = 1, \ldots, L$ and $l \neq i$. For a transformed test vector $\mathbf{z}^*_{n,b}$ using the VAE's encoder, its similarity to $\mathbf{z}^0_{n,b}$ is u_0, and to $\mathbf{z}^1_{n,b}$, u_1. With a threshold δ, if $u_0 - u_1 > \delta$, $\mathbf{z}^*$ is assigned to cluster H_0; otherwise, it is assigned to H_1. The FeRAP framework is detailed in Algorithm 4.

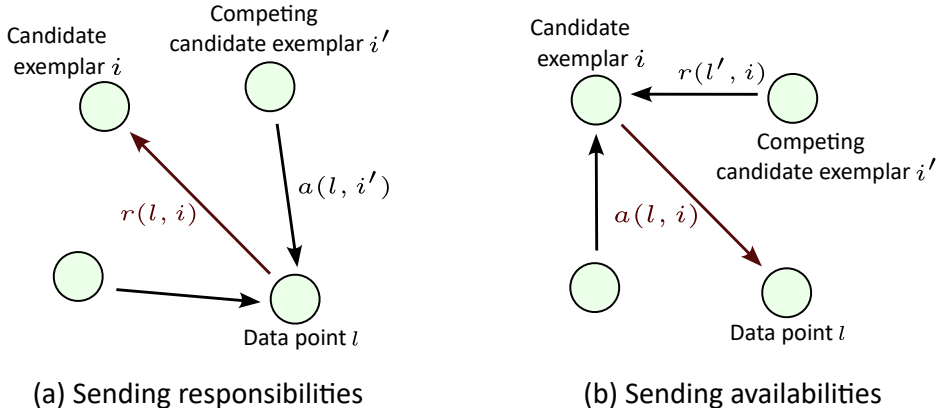


Figure 5.3: Messages passed during training of the Affinity Propagation algorithm (a) "Responsibility" $r(l, i)$ is the message sent from data points to candidate exemplars, representing how strongly a data point prefers a particular candidate exemplar over others. (b) "Availability" $a(l, i)$ is the message sent from candidate exemplars to data points, reflecting the extent to which a candidate exemplar is suitable to serve as a cluster center for a given data point.

Algorithm 4 The proposed FeRAP approach for cooperative spectrum sensing in mobile large-scale networks.

Mobile SUs are deployed to collect sensing data $\mathbf{Y}_{n,b}$
Initialize each SU's VAE model parameters ϕ_n and θ_n.
Set iteration counter $t = 0$
while $t \leq$ MaxEpoch **do**
    SUs train their VAE model and send their updated parameters to FC.
    Aggregate the VAE's model parameters at FC.
    FC broadcasts the final VAE's model parameters ϕ^* and θ^* back to the SUs.
    $t = t + 1$
end while
SUs utilize the collaboratively learned VAE parameters to map $\mathbf{Y}_{n,b}$ onto the encoder's latent space, generating $\mathbf{Z}_{n,b}$.
SUs independently train the AP algorithm on $\mathbf{Z}_{n,b}$ to determine the number of clusters and exemplars.
Determine the state of the primary channel.

5.6 Simulation Results

In this section, we evaluate the performance of our proposed FeRAP approach, which leverages deep federated representation learning for CSS in large-scale mobile CR networks.

5.6.1 Setup

We consider a large-scale CR network covering a $6 \times 6\,\text{km}^2$ area, consisting of $n = 4$ SUs moving along straight paths, each path containing $L = 80$ data points. Notably, FeRAP's performance remains independent of the specific routes, allowing SU paths to be tailored according to the geographical environment in real-world applications. The sensing duration for each SU is set to $\tau = 100\,\mu$s. The primary network includes $m = 2$ PUs. There are $b = 2$ channels, each with a bandwidth of $\omega = 5$ MHz, with two SUs sensing channel 1 and the other two sensing channel 2 as they traverse the area. The PUs transmit independently with a power of $\rho_m = 250$ mW. Importantly, the mobile SUs do not have access to the transmission power information of the PUs. The noise power spectral density is $\eta = -174$ dBm. Initially, the path loss exponent is set to $\alpha = 4$, and the Nakagami-ν shape factor is set to $\nu = 1$, after which these parameters are adjusted to evaluate the system's performance under different propagation environments and fading conditions. The shadow fading component $\psi_{m,n}$ is assumed to be quasi-static during sensing, and the VAE loss parameter β

in (5.12) is set to 1.5.

To evaluate our unsupervised FeRAP approach for CSS, we compare it to two supervised DL methods: a standalone, fully connected DNN and a FL model with a fully connected DNN at each SU n. Additionally, we compare FeRAP to the vanilla unsupervised AP algorithm. The proposed VAE architecture consists of 6 layers with 80, 3, 2×2, 2, 3, and 80 neurons, respectively. The fully connected DNN has 3 layers with 80, 2, and 1 neurons. However, it is important to note that generating labeled data (i.e., sensing data and the corresponding label H_0/H_1) for supervised learning is impractical in realistic CR networks. All models are trained with an Adam optimizer (learning rate $\eta = 10^{-3}$) for 120 epochs, using batches of 100 energy vectors. The dataset of each SU is split into training, validation, and testing sets with 2400, 1000, and 10,000 samples, respectively. The sensing performance is evaluated using the Receiver Operating Characteristics (ROC) and the Area Under the ROC Curve (AUC).

5.6.2 Results and Analysis

Fig. 5.4 compares the detection performance of the unsupervised FeRAP approach with that of the supervised standalone and federated DNNs. It also evaluates the performance of the vanilla AP algorithm trained on high-dimensional sensing data against FeRAP. The results in Fig. 5.4 reveal a significant performance gap between the supervised DNN and the AP algorithm, as the unsupervised AP fails to form a reliable clustering model in high-dimensional data. In contrast, FeRAP, which

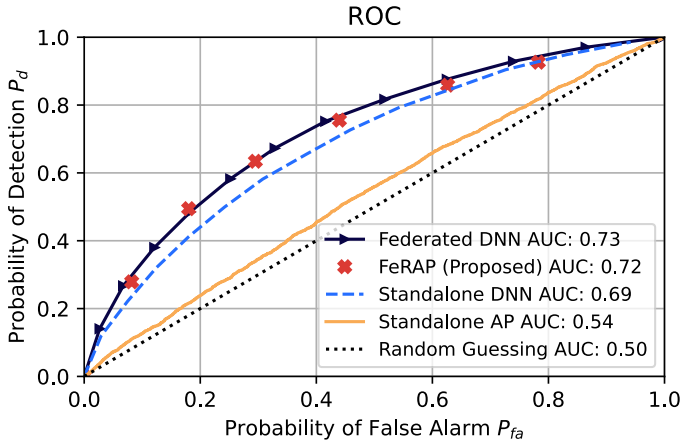


Figure 5.4: Benchmarking the detection performance of our proposed FeRAP approach.

utilizes a VAE to learn a low-dimensional latent representation of the sensing data, significantly outperforms the vanilla AP. Moreover, FeRAP outperforms the standalone DNN without the need for labeled data, prior knowledge, or transmitting private sensing data to the FC. Impressively, it achieves performance similar to the supervised federated DNN by leveraging joint model training and the spatial diversity of the SUs.

Fig. 5.5 examines the sensing performance of the FeRAP approach at different levels of PU transmission power ρ_m. Wireless standards, such as IEEE 802.11, GSM, and LTE, define various transmission power settings to adapt to changing channel conditions [11], making it essential to assess FeRAP's detection probability P_d over a range of ρ_m values. The figure demonstrates a positive correlation between detection probability P_d and increasing transmission power ρ_m; as ρ_m increases, the spectral energy rises, enabling the SUs to better distinguish between idle spectrum (with only noise) and occupied spectrum. Notably, FeRAP's detection performance closely matches that of the supervised federated DNN across all ρ_m values, in contrast to the vanilla AP. Furthermore, FeRAP outperforms the standalone DNN in detection performance.

To assess FeRAP under various propagation conditions, we modify the path loss exponent α. Specifically, we set $\alpha = 2.42$ for outdoor line-of-sight (OL) environments, select $3.5 < \alpha \leq 4$ for non-line-of-sight (NLOS) conditions, and choose $\alpha = 4.5$ for obstructed environments. As illustrated in Fig. 5.6, detection performance declines with increasing α, with obstructed environments presenting the most difficult conditions. However, FeRAP maintains strong performance in both OL

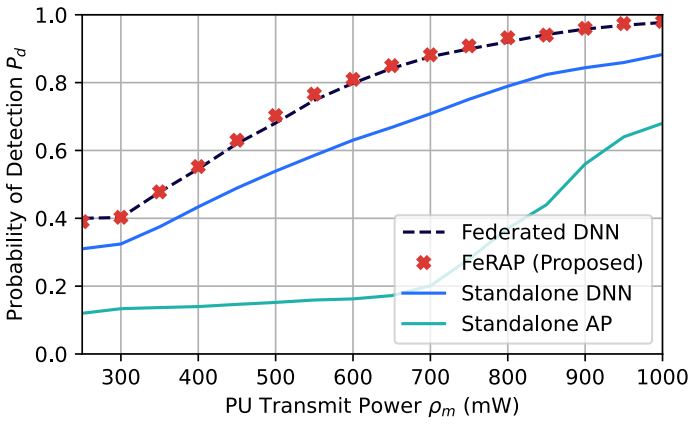


Figure 5.5: Effect of ρ_m on detection performance.

and NLOS conditions. To further evaluate FeRAP under fading effects, we keep $\alpha = 4$ constant and adjust the fading severity ν. As shown in Fig. 5.7, higher values of ν reduce the impact of fading, improving sensing performance, while lower values of ν increase fading effects, leading to decreased performance. Despite the challenges posed by fading, FeRAP consistently delivers reliable sensing performance, thanks to the VAE's role in both preprocessing and feature extraction, along with the collaborative federated training, which allows SUs to effectively mitigate the adverse effects of fading.

To assess the scalability of our FeRAP approach for CSS, we vary the number of cooperating SUs n and the number of PUs m. The results in Fig. 5.8 clearly show that as the number of SUs n involved in the joint FeRAP training increases, the sensing performance improves significantly.

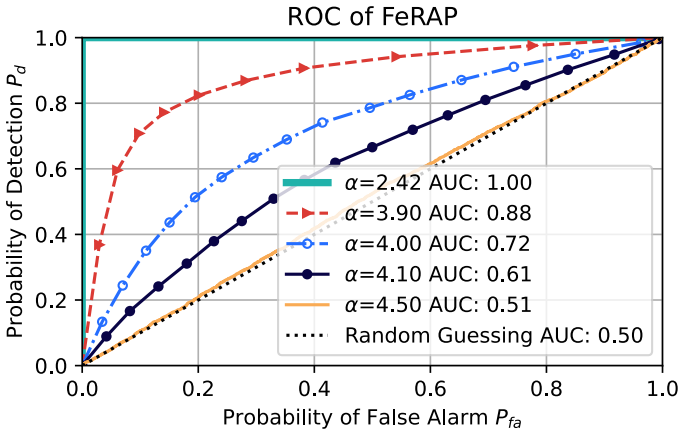


Figure 5.6: Effect of α on detection performance of FeRAP.

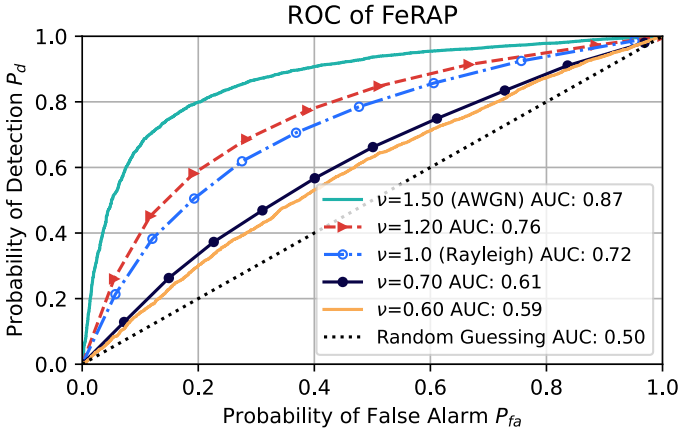


Figure 5.7: Effect of Nakagami-m fading parameter on detection performance of FeRAP.

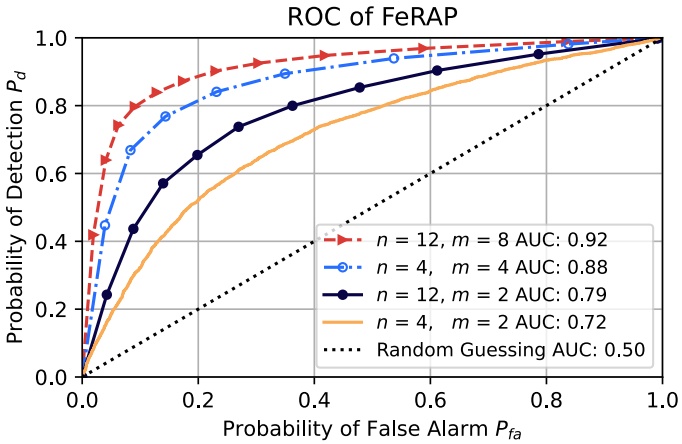


Figure 5.8: The detection performance of FeRAP under various n SUs and m PUs.

This indicates that the collective participation of SUs strengthens FeRAP's ability to accurately detect spectrum usage. Additionally, increasing the number of PUs m utilizing the spectrum, while keeping the number of SUs constant, leads to a noticeable improvement in detection performance. This enhancement is likely due to the greater ability of the SUs to differentiate between unoccupied and occupied spectrum bands. These results underscore FeRAP's scalability and its potential to support large-scale CR networks with an increasing number of devices.

5.7 Conclusions

In this chapter, we introduced FeRAP, the first unsupervised deep federated representation learning approach for cooperative sensing in large-scale mobile CR networks. FeRAP offers several key advantages over existing DL-based CSS methods. First, its distributed learning approach significantly reduces the communication overhead seen in centralized CSS techniques, which require SUs to send their sensing data to a fusion center. Additionally, FeRAP enhances sensing performance by capitalizing on cooperation among spatially diverse SUs. Second, this distributed framework ensures sensing data privacy, placing control back into the hands of the SUs. Thirdly, this framework enables a newly joined SU to download the trained β-VAE model parameters, allowing it to transform spectrum data for analysis and decision-making. Moreover, these downloaded parameters also allow the SUs to generate new synthetic samples, as the β-VAE is a generative model. FeRAP is a

fully data-driven solution, requiring no prior knowledge, and can be trained using locally collected data. Our results demonstrate that FeRAP outperforms traditional non-deep learning methods while achieving performance comparable to supervised federated learning. Additionally, we showcased its scalability across different node deployments and its robustness in diverse network configurations, propagation environments, and fading conditions.

Chapter 6

Towards Self-Managing and Sustainable Spectrum Sharing Networks

6.1 Introduction

The rise of the Internet of Things (IoT) has significantly transformed how we interact with the world. As the number of interconnected devices increases, the IoT has opened up new opportunities for remote monitoring, automation, and the creation of intelligent urban environments. However, the ongoing expansion of devices and services, along with the growing demand for spectrum resources, has shifted the focus toward enhancing spectral efficiency [76]. Additionally, IoT devices must coexist with other technologies like Bluetooth and Wi-Fi [77], leading to unavoidable spectrum congestion within the allocated IoT bands. To address these issues, CIoT, a combination of CR technology and IoT networks, has emerged as a solution for optimizing spectrum utilization efficiency [18, 60]. Using the underlay CR approach, CIoT devices operate as SUs that are permitted to use spectrum resources allocated to PUs, provided their transmissions do not interfere with PUs communications [41, 78]. Furthermore, CIoT devices compete with each other due to their varied requirements, ranging from ultra-reliable, low-latency communications (URLLCs) to maximizing QoS [79]. This diversity introduces challenges related to fairness and interference. Additionally, interference in CIoT networks can arise from both PUs and SUs, given their shared use of the channel. This may result in a degradation of signal quality and a reduction in overall performance. Consequently,

effective channel access schemes are crucial for ensuring efficient spectrum sharing.

The main challenge in underlay CIoT is to develop a dynamic power control strategy that enables secondary devices to adjust their transmit power while remaining below the PUs' interference threshold and maximizing their throughput [80, 81]. Power control strategies in CR networks are classified into cooperative and non-cooperative categories. Cooperative strategies involve SUs collaborating to optimize power levels for a shared goal, while non-cooperative strategies have SUs independently determining their power levels without considering the overall network performance. Power control becomes especially complex for energy-constrained CIoT devices, which must also prioritize extending the network's lifetime. Energy Harvesting (EH) is an emerging solution to enhance the sustainability of energy-limited CIoT networks, as radio frequency EH enables CIoT devices to harvest energy from radio frequency sources generated by nearby devices [82].

6.2 Related Works

Reinforcement Learning (RL) has recently gained significant attention for its ability to navigate complex and dynamic environments, such as those encountered in CIoT, without the need for detailed or accurate prior knowledge [18]. Several studies have investigated Deep Reinforcement Learning (DRL), which integrates RL with Deep Learning (DL), to devise power control strategies. However, centralized DRL methods, like those in [36,83,84], often face issues related to convergence and security. While distributed power control strategies in CIoT networks have been explored in works such as [79, 85, 86], they mostly rely on Multi-Agent Reinforcement Learning (MARL). Although MARL enables collaborative learning through the exchange of state information, it brings about convergence challenges and increases signaling overhead. As the number of agents grows, maintaining stable and efficient multi-agent power control becomes more difficult. Moreover, MARL approaches require substantial communication and resource utilization, which makes them less feasible for energy-limited devices.

To address these challenges, recent studies have moved towards RL and DRL-based non-cooperative power control strategies. In [80], a convolutional Deep Q-Network (DQN) was employed for power control in full-duplex CR, successfully meeting QoS and interference constraints. [6]

proposed a power control strategy using a DQN, considering factors such as channel occupancy, channel gain, and energy arrival to optimize the total achievable rate of an EH-enabled CR device. A DQN-based power control strategy for an SU transmitter was suggested in [87], which adhered to interference and energy constraints while employing a Time Switching (TS) protocol to switch between EH and data transmission. The study in [88] introduced a power control approach based on a DQN within a CR network with a single SU and a primary network consisting of one PU.

CIoT devices, which may be owned by various organizations with different objectives, can have conflicting goals, leading to competition for spectrum access. A non-cooperative game for power control is presented in [89], where dynamic learning is used to optimize power allocation for multiple SUs with minimal overhead. However, the authors assume a simplistic feedback mechanism from the primary user base station, which is unrealistic due to the lack of cooperation between primary and secondary networks [9, 10]. In [90], the power control problem in CR networks was addressed using an Asynchronous Advantage Actor-Critic (A3C) DRL approach, implemented by multiple competing SUs in parallel. The authors of [91] proposed a Q-learning-based distributed power control method for multiple SUs, aiming to maximize energy efficiency while respecting QoS and interference constraints. Although effective, Q-learning faces scalability issues, particularly in large state spaces. Additionally, Q-learning's inability to generalize across similar contexts limits its adaptability to evolving environments.

Based on the analysis of the aforementioned works, studies such as [6, 80, 87] have largely focused on developing non-cooperative power control strategies, often overlooking the interactions among different SUs. It is important to recognize that CIoT devices often have conflicting interests. In pursuit of higher transmission rates, each device naturally aims to optimize its transmit power. However, this individualistic approach can lead to increased interference levels. Furthermore, works like [6, 87] make the unrealistic assumption of a consistently stable radio signal source for EH. While [89, 90] acknowledge the presence of competing SUs, they fail to consider their energy constraints. Additionally, the assumption made in [89] regarding PU-SU communication is not feasible in underlay CR environments [9, 10]. Moreover, [90] overlooks the varied capabilities of learners within the CR network, assuming uniformity across devices. However, CIoT networks typically involve a wide range of devices with differing capabilities [18]. These gaps highlight the

urgent need for a new approach to joint power control and channel access coordination, specifically designed for energy-constrained CIoT networks.

Most studies that apply DRL to optimize transmission in CRs have primarily concentrated on managing either power [36, 80] or time allocation [92] individually. Additionally, they overlook the potential of an integrated approach that could significantly improve both data transmission efficiency and EH. While [93] proposed a joint strategy to optimize both time and power, they took a multi-agent Device-to-Device (D2D) perspective, where CRs not only share common objectives but can also harvest energy from each other's transmissions. Likewise, [94] explored the joint optimization of time and power allocation, but their approach relies on offline optimization methods that require prior knowledge and assumptions about the radio environment. [87] developed a DRL-based power control strategy for EH-enabled CR networks, but it treats time allocation for CR activities as a fixed hyperparameter. This approach not only requires prior knowledge for optimal performance but also limits adaptability to changes in the environment. This highlights the need for more dynamic and flexible learning approaches that can not only manage power more effectively but also optimize time allocation in the ever-changing CIoT environments.

6.3 Contributions

Based on the identified gaps in the literature and the pressing need for innovative solutions in the field of CIoT networks, our contributions focus on two critical aspects. First, we address the challenge of developing an intelligent strategy that effectively manages and optimizes joint power control and channel access coordination. Second, we introduce a dynamic learning approach that efficiently manages power and time allocation in dynamic CIoT scenarios. We summarize our contributions below:

- We first introduce a system model for a Wireless Power Transfer (WPT)-enabled CIoT network with multiple competing users, where a battery-operated CIoT Tx must intelligently manage its transmit power and decide whether to harvest energy or transmit data. We define the optimization problem to maximize the long-term achievable sum rate of the CIoT network, considering channel occupancy, competition, channel gain, energy arrival, battery capacity,

and interference constraints. This formulation guides the coordination of joint power control and channel access within the CIoT network, modelled as a Markov Decision Process (MDP) without cooperation between devices for state information.

- Subsequently, we introduce a novel Deep Q-Network (DQN)-based DRL algorithm that features an innovative non-linear activation function designed to combat the "dying neuron" problem, facilitating quicker convergence and improving the overall dynamics of the learning process. To further enhance stability and expedite convergence, we propose a novel initialization method based on Kaiming (He) [95] for setting the DQN's parameters. Additionally, we present a thoughtfully designed scheduler that dynamically adjusts the learning rate (gradient update rate) based on the model's performance.

- We introduce a system model for a Simultaneous Wireless Information and Power Transfer (SWIPT)-enabled CIoT network, where the network gains greater flexibility by controlling both its transmit power and the time allocated to each activity (EH and transmission). Furthermore, we adopt a more realistic EH approach, enabling the agent to recharge from ambient sources without relying on a dedicated stable source. We define the optimization problem to maximize the long-term achievable sum rate while accounting for channel occupancy, channel conditions, energy arrival, battery capacity, and interference constraints. The problem is then modelled as a discrete-time model-free MDP with continuous states and discrete actions.

- To intelligently manage the allocation of time between EH and transmission, we propose a novel lightweight Double Deep Q-Network (DDQN) designed to autonomously learn an operation policy for the CIoT agent. Furthermore, the Double Deep Q-Network (DDQN) also allows the CIoT agent to dynamically adjust its transmit power to optimize both long-term achievable throughput and network lifetime, considering factors such as channel occupancy, energy arrival patterns, and interference constraints. Additionally, we introduce an Upper Confidence Bound (UCB) strategy that effectively optimizes decision-making in the CIoT environment.

- We evaluate the performance of the proposed DRL algorithms and benchmark them against

other works in the literature. Extensive simulations demonstrate the effectiveness of the proposed algorithms, showcasing their ability to converge to a stable state across various simulation settings. Moreover, in terms of average sum rate, average achievable reward, and interference ratio, the proposed algorithms significantly outperform existing baseline approaches.

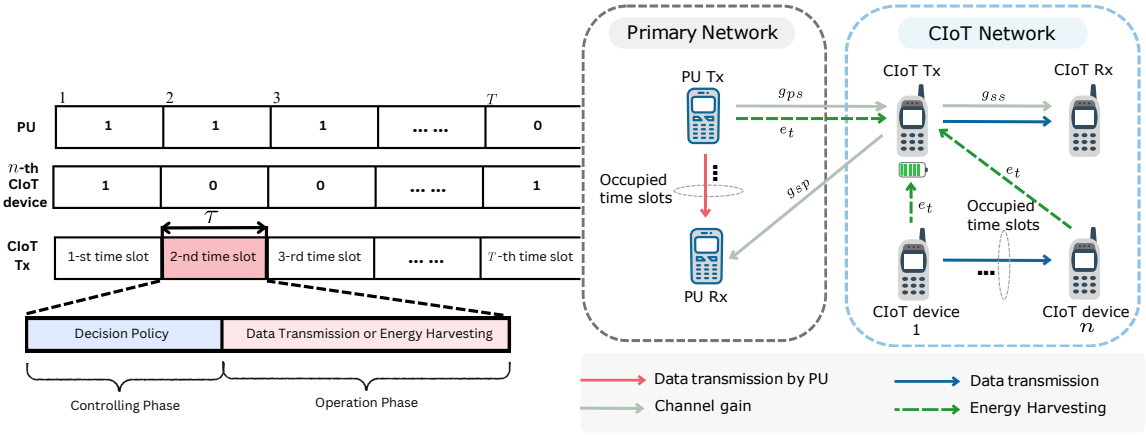
6.4 Joint Power Control and Access Coordination in WPT-EH CIoT

6.4.1 System Model and Problem Formulation

Cognitive IoT System

Consider the time-slotted communication system depicted in Fig. 6.1 (a), which operates over a finite set of time slots indexed by $t = 1, ..., T$, each of duration τ. Each time slot t is divided into two distinct phases: a controlling phase and an operation phase. In the controlling phase, a CIoT device must sense the presence of PUs and decide whether to transmit data or engage in energy harvesting for that time slot. Furthermore, the device must determine its transmit power according to a strategy. In the operation phase, the CIoT device carries out either its data transmission or energy harvesting process.

In an underlay CIoT network, a secondary Transmitter-Receiver (Tx-Rx) pair shares the spectrum allocated to the primary network. The system model under consideration is illustrated in Fig. 6.1 (b). There are N additional CIoT devices indexed by $n = 1, .., N$, with whom the CIoT Tx must coordinate channel access. We adopt a model where only one CIoT device can use a time slot at any given moment. This structure ensures that when a single CIoT device occupies multiple time slots, it is equivalent to multiple CIoT devices, each occupying one slot. This design enables the system to scale effectively as the number of CIoT devices grows. The CIoT Tx is equipped with a rechargeable battery of capacity B_{max} and is capable of Wireless Power Transfer (WPT) EH. The number of time slots used by PUs for transmission is denoted by L, where $1 < L < T$, and a PU Tx transmits consistently at a power level of P_p^t in all L slots. In the t-th time slot, if the PU Tx is



(a) Time-slotted operation.

(b) System model.

Figure 6.1: Illustration of the designed WPT-enabled CIoT network: (a) its time-slotted operation and (b) its system model.

active, the PU status indicator ω_p^t is set to 1; otherwise, it is set to 0. That is,

$$\omega_p^t = \begin{cases} 1 & \text{if the PU Tx is using time slot } t, \\ 0 & \text{otherwise.} \end{cases} \tag{6.1}$$

Similarly, if the n-th CIoT device is transmitting data in the t-slot, the status indicator ω_n^t is then 0, otherwise it is 1. Consequently,

$$\omega_n^t = \begin{cases} 0 & \text{if the } n\text{-CIoT device is using time slot } t, \\ 1 & \text{otherwise.} \end{cases} \tag{6.2}$$

In underlay CR, a CIoT Tx can use the same slot as a PU Tx as long as it adheres to the interference threshold I_{th}. That is, the CIoT Tx needs to adjust its transmit power P_s^t such that

$$P_s^t g_{sp}^t \leq I_{th}, \tag{6.3}$$

where g_{sp}^t represents the channel power gain between the CIoT Tx and the PU Rx. However, CIoT devices must coordinate their transmissions to avoid interference. Specifically, if the n-th CIoT device is using the t-th time slot, i.e., $\omega_n^t = 0$, the CIoT Tx should refrain from transmitting data.

The channel power gains for the CIoT Tx-Rx pair g_{ss}^t, the PU Tx and CIoT Rx pair g_{ps}^t, and the CIoT Tx and PU Rx pair g_{sp}^t, are modeled as independently and identically distributed (i.i.d.) Rayleigh fading channels. These channel power gains are assumed to remain constant within a time slot, but may vary independently across different time slots.

We represent the CIoT device's decision between transmission and energy harvesting as d_t. If $d_t = 0$, the CIoT device engages in data transmission mode; conversely, if $d_t = 1$, the CIoT device harvests energy. Accordingly,

$$d_t = \begin{cases} 0 & \text{Data transmission mode in time slot } t, \\ 1 & \text{Energy harvesting mode in time slot } t. \end{cases} \tag{6.4}$$

When the t-th time slot is idle, the achievable rate of the CIoT Tx is

$$R_0^t = \log_2\left(1 + \frac{P_s^t g_{ss}^t}{\sigma^2}\right), \tag{6.5}$$

where P_s^t is the transmit power of the CIoT Tx, and σ^2 is the channel noise variance. If the channel is occupied by a PU Tx at the t-th time slot, the achievable rate of the CIoT Tx decreases due to the interference of the PU, which is given by

$$R_1^t = \log_2\left(1 + \frac{P_s^t g_{ss}^t}{P_p^t g_{ps}^t + \sigma^2}\right). \tag{6.6}$$

Therefore, the achievable rate of the CIoT Tx at a time slot t can be written as

$$R^t = \omega_n^t (1 - d_t)\left[(1 - \omega_p^t) R_0^t + \omega_p^t R_1^t\right]. \tag{6.7}$$

Energy Harvesting Model

The Energy Harvesting (EH) process is modeled as an energy arrival process, where energy is harvested in independent and identically distributed time slots. The energy harvested at time $t = 0$ is set to $e_0 = 0$. It is assumed that the energy harvested during each time slot e_t follows a uniform distribution: $e_t \sim U(0, E_{max})$. The value of E_{max} depends on the radio signals, which are

influenced by the presence of the other N CIoT devices and the PU Tx in each time slot. Therefore, E_{max} is

$$E_{max} = \begin{cases} P_p^t & \text{if } \omega_p^t = 1, \text{ PU transmitting,} \\ P_n^t & \text{if } \omega_n^t = 0, \text{ CIoT } n \text{ transmitting,} \\ P_p^t + P_n^t & \text{if } \omega_p^t = 1 \text{ and } \omega_n^t = 0. \end{cases} \tag{6.8}$$

where P_n^t is the transmit power of the n-th CIoT device. It should be noted that the EH process does not result in any further increase in energy consumption for PUs or other CIoT devices [86].

The initial battery level of the CIoT transmitter is denoted as B_0, and B_t represents the available battery energy at the t-th time slot. Following the assumption in [84], we consider the rechargeable battery to be ideal, meaning there are no energy losses during storage or retrieval. Energy consumption in CIoT devices is solely due to data transmission. Furthermore, any excess harvested energy that exceeds the battery's full capacity is considered discarded. Time slots are normalized, allowing for the interchangeable use of the terms "energy" and "power" [84]. At any given time t, the transmit power P_s^t selected by the CIoT Tx must not exceed the total energy available in the battery, B_t. To this end,

$$0 \leq (1 - d_t) P_s^t \tau \leq B_t, \tag{6.9}$$

where τ is the duration of the time slot. In each time slot t, the battery's energy storage or usage depends on the choice between energy harvesting and data transmission. At the next time slot, $t+1$, the available energy is updated based on the decision d_t made by the CIoT device, as follows

$$B_{t+1} = \min\{B_t + d_t e_t - (1 - d_t) P_s^t \tau, B_{max}\}. \tag{6.10}$$

Problem Formulation

Our goal is to develop a dynamic algorithm for joint power control and channel access coordination in the CIoT network, aiming to ensure continuous energy availability while improving the likelihood of successful data transmissions. The primary challenge is to maximize the sum rate of the CIoT network by optimizing both transmit power P_s^t and the decision d_t. This optimization process considers factors such as the interference threshold I_{th}, coordination of channel access with

N other CIoT devices, the available battery energy B_t, and the harvested energy e_t. Therefore, the task of maximizing the sum rate for the CIoT transmitter is formulated as a constrained optimization problem, expressed as

$$\max_{d_t, P_s^t} \sum_{t=1}^{T} \omega_n^t (1 - d_t) \left[(1 - \omega_p^t) R_0^t + \omega_p^t R_1^t \right] \tag{6.11a}$$

$$\text{s.t.} \sum_{t=1}^{k} P_s^t \tau \leq B_0 + \sum_{t=0}^{k-1} e_t, \ \ \forall k, \tag{6.11b}$$

$$0 \leq (1 - d_t) P_s^t \tau \leq B_t, \ \ d_t \in I \triangleq \{0, 1\}, \tag{6.11c}$$

$$d_t = 1, \ \ \forall \omega_n^t = 0 \text{ for } n = 1, ..., N, \tag{6.11d}$$

$$\omega_p^t g_{sp}^t P_s^t \leq I_{th}, \ \ \omega_p^t \in \Omega \triangleq \{0, 1\}, \tag{6.11e}$$

where k denotes the number of slots the CIoT Tx decides to transmit. (6.11b) ensures that the transmission power P_s^t stays within the bounds of the initial battery energy B_0 and the cumulative harvested energy across all time slots within the frame. (6.11c) requires that the CIoT Tx transmits with a maximum power level equal to the current battery energy B_t during any given time slot t. (6.11d) prevents the usage of time slots t that are concurrently used by other competing CIoT devices. Finally, (6.11e) ensures that the CIoT transmitter adheres to the interference threshold I_{th} to avoid causing harmful interference to the PU in every time slot t.

To achieve the optimization objective in (6.11a), the CIoT Tx must learn to make intelligent decisions regarding both d_t and P_s^t by leveraging locally observable information such as PU and CIoT activity, remaining energy, and channel quality. If the CIoT Tx had prior knowledge of such factors, the problem could be addressed using classical offline methods [36]. However, by implementing a Deep Reinforcement Learning (DRL) algorithm, the CIoT Tx can optimize its decision-making process effectively without requiring prior knowledge about the environment.

We model the decision-making process of the CIoT Tx as a stochastic control process within a discrete-time system. The Markov property indicates that the next state of a system depends only on its current state and action, not on any previous states. In our system, the stochastic behavior of states, including the activities of the PU and CIoT devices, as well as energy arrivals, adheres to the

Markov property. Consequently, the problem of learning the optimal strategy for power control and channel access can be effectively formulated as a discrete-time Markov Decision Process (MDP) with continuous states and discrete actions. The MDP tuple is defined as $(\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{P}}, \boldsymbol{\mathcal{R}}, T)$, where $\boldsymbol{\mathcal{S}}$ represents the set of states of the environment, $\boldsymbol{\mathcal{A}}$ denotes the set of actions, $\boldsymbol{\mathcal{P}}$ is the set of state transition probabilities, $\boldsymbol{\mathcal{R}}$ is the set of rewards associated with the state-action pairs, and T represents the time step.

In practical scenarios, obtaining the exact Probability Density Function (PDF) of energy and channel fading is challenging, making it difficult to acquire the exact state transition probabilities $\boldsymbol{\mathcal{P}}$ [6]. As a result, we adopt a model-free MDP and develop a DRL framework to estimate $\boldsymbol{\mathcal{R}}$ given $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{A}}$ without needing $\boldsymbol{\mathcal{P}}$. In this model, the CIoT agent [1] learns a policy $\pi : \boldsymbol{\mathcal{S}} \rightarrow \boldsymbol{\mathcal{A}}$ through continuous exploration and training with the environment, which maximizes the accumulated reward. Therefore, the model-free MDP tuple becomes $(\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{R}}, T)$. Below, we define the components of the MDP tuple.

State Space $\boldsymbol{\mathcal{S}}$: In each time slot, the CIoT Tx, as a learning agent, observes the state of the environment and uses this information for decision-making. The state space consists of all states across T time slots. At each time s_t, the agent considers factors such as the current battery level B_t, the energy harvested in the previous time slot e_{t-1}, whether the slot is occupied by a PU Tx or any of the N other CIoT devices, and the channel power gains $g_{ps}^t, g_{sp}^t, g_{ss}^t$. Therefore, the state at the t-th time slot is represented by

$$\begin{gathered} s_t = \{B_t, e_{t-1}, \omega_p^t, \omega_n^t, g_{ps}^t, g_{sp}^t, g_{ss}^t\}, \\ \text{for } n = 1, ..., N. \end{gathered} \tag{6.12}$$

The environment in (6.12) is defined by the occupancy status of the N CIoT devices, allowing it to accommodate a diverse range of devices, each with different capabilities. The occupancy state remains independent of the specific capabilities of the devices, which highlights the flexibility of CIoT networks.

Action Space $\boldsymbol{\mathcal{A}}$: The action space consists of all possible actions that the CIoT agent can take. Based on the state of the environment s_t, the CIoT agent must decide whether to transmit data

[1] The term CIoT agent refers to the CIoT Tx within the examined network.

($d_t = 0$) or harvest energy ($d_t = 1$). Additionally, it must determine the appropriate transmit power P_s^t. Thus, the action of the CIoT agent at the t-th time slot is represented as $a_t = [d_t, P_s^t]$, where $d_t \in I \triangleq \{0, 1\}$ and $P_s^t \in P$.

Reward $\mathcal{R}$: The CIoT agent evaluates the quality of its chosen action based on the reward it receives, which helps refine its decision-making strategy. As such, we consider the achievable rate as the reward when the CIoT Tx transmits data and adheres to the constraints in (6.11). If the CIoT Tx opts to harvest energy, the reward is 0. If the CIoT Tx selects an action a_t that violates the constraints in (6.11), a negative reward is assigned as a penalty. Therefore, the reward r_t for the CIoT agent at each time slot t is expressed as

$$r_t = \begin{cases} R_0^t & d_t = 0, \omega_p^t = 0, \omega_n^t = 1, 0 \leq P_s^t \tau \leq B_t, \\ R_1^t & d_t = 0, \omega_p^t = 1, \omega_n^t = 1, 0 \leq P_s^t \tau \leq B_t, P_s^t g_{sp}^t \leq I_{th}, \\ 0 & d_t = 1, P_s^t \tau > B_t, \\ -\phi & \text{others.} \end{cases} \tag{6.13}$$

Time Step T: We define each transition from time slot t to $t + 1$ as a single step. We iterate through each state-action pair for all time slots T.

6.4.2 Deep Q-Network with ϵ-Greedy Exploration Strategy

In the model-free MDP, the CIoT agent faces the challenge of determining the state-action value without prior knowledge of $\mathcal{P}$. However, by employing Reinforcement Learning (RL), the CIoT agent can approximate the state-value function and develop a strategy π to select actions based on the current state of the environment. The goal is to maximize the long-term cumulative reward (rate) using π, while adhering to the constraints of the CIoT system. Q-learning, an RL algorithm, is used to estimate the expected state-action value function, known as the Q-function. The Q-function can be expressed as

$$Q^\pi(s, a) = \mathbb{E}[r_t + \gamma \max_a Q^\pi(s_{t+1}, a) | s_t = s, a_t = a], \tag{6.14}$$

where r_t represents the immediate reward for a given action a_t and state s_t. The term $\gamma \max_a Q^\pi(s_{t+1}, a)$ represents the discounted expected future reward, with $\gamma \in \{0, 1\}$ being the discount factor. The

value of γ controls the weight of future rewards compared to immediate rewards, with larger values emphasizing long-term rewards. The objective of the CIoT agent is to determine the optimal action a that maximizes the Q-value at each time slot t.

The Proposed Deep Q -Network Architecture

Q-learning can sometimes experience slow convergence when trying to identify the best actions to resolve a problem [85]. As a result, we utilize a Deep Q-Network (DQN), an enhanced version of Q-learning that uses a Deep Neural Network (DNN) to approximate the Q-function. The DQN predicts the cumulative reward (i.e., Q-value) for each possible action a in a specific state s. In other words, the DQN adjusts its parameters $\boldsymbol{\theta}$ to ensure that

$$Q^{\pi}(s, a; \boldsymbol{\theta}) \approx Q^{\pi}(s, a). \tag{6.15}$$

The DQN is implemented using a fully connected DNN. The input layer of the DQN contains j neurons, which correspond to the dimensionality of the state space $\boldsymbol{\mathcal{S}}$. The DQN also includes two hidden layers, with h_1 and h_2 neurons in each layer, respectively. The output layer consists of z neurons. The parameters $\boldsymbol{\theta} = \{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}$ for $i = \{1, \ldots, 4\}$, represent the weights and biases of the DQN network layers.

To initialize the weights of the DQN, we use Kaiming (He) initialization [95]. This approach initializes the weights by sampling from a Gaussian distribution $\mathcal{N}(0, \frac{2}{\nu_i})$, where ν_i represents the number of input neurons in each layer i. By applying He initialization, faster convergence is encouraged, and generalization during training is improved. Additionally, a leaky Rectified Linear Unit (ReLU) activation function $f(.)$ is used within the Q-network architecture. The leaky ReLU activation helps prevent the "dying ReLU" problem (where neurons output zero) by ensuring that all neurons contribute to the learning process, which leads to faster convergence and enhanced learning dynamics. The leaky ReLU activation is defined as

$$f(x) = \begin{cases} x, & \text{if } x \geq 0, \\ \alpha x, & \text{if } x < 0, \end{cases} \tag{6.16}$$

where $\alpha \in \{0, 1\}$ is the "slope" used to adjust the "leakiness" of the ReLU.

During training, a Target DQN is utilized, initially identical to the DQN. However, as training progresses, the parameters of the Target DQN are updated less frequently than the DQN, typically spanning multiple training steps. The update rate of the Target DQN is denoted as κ. To train the proposed DQN, the Mean Squared Error (MSE) loss $\mathcal{L}$ is employed to compute the MSE between the predicted Q-values and the target Q-values as

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}\left[\left[(\mathbf{r}_t + \gamma \arg\max_{\mathbf{a}\in\boldsymbol{\mathcal{A}}} Q^{\pi}(\mathbf{s}_{t+1}, \mathbf{a}; \boldsymbol{\theta}')) - Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t; \boldsymbol{\theta})\right]^2\right], \tag{6.17}$$

where $\boldsymbol{\theta}'$ represents the parameters of the target DQN. The DQN is trained using experience replay [83]. Specifically, an experience replay buffer of size m is utilized to store past experiences (s_t, a_t, r_t, s_{t+1}) in the memory $\mathcal{M}$. When the memory reaches its capacity, the dataset of state-action pairs is sampled into mini-batches of experiences, which are then used during training to update the parameters of the DQN. This approach helps mitigate temporal correlations in the data, reducing the potential for training instability.

During training, the goal is to minimize the loss in (6.17) over a mini-batch of state-action pairs $(\mathbf{s}, \mathbf{a})$. That is,

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathbf{s}, \mathbf{a}). \tag{6.18}$$

The backpropagation algorithm [68] can be used effectively to calculate $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}; \mathbf{s}, \mathbf{a})$, which is the gradient of the loss with respect to the DQN's parameters given the state-action pairs. Using the obtained $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}; \mathbf{s}, \mathbf{a})$, Stochastic Gradient Descent (SGD) can be used to update the parameters of the DQN accordingly as

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}; \mathbf{s}, \mathbf{a}), \tag{6.19}$$

where $\eta \in \{0, 1\}$ is the learning rate, which controls the step size at each iteration of SGD. Additionally, a well-designed learning rate scheduler is utilized. It starts with an initial learning rate that guarantees stable learning in the early stages. As training progresses, the scheduler adapts the learning rate dynamically, taking into account factors such as model performance and a defined patience period. This approach helps achieve efficient convergence and enhances overall

performance.

ϵ-Greedy Exploration Strategy

To investigate the environment and identify the best strategies, we utilize an ϵ-greedy exploration method. This strategy effectively manages the balance between exploitation and exploration. In the ϵ-greedy method, the CIoT agent selects the action that maximizes the estimated Q-value (exploitation) with probability $1-\epsilon$, while opting for a random action (exploration) with probability ϵ. Lower values of ϵ encourage exploitation, whereas higher values promote exploration. To enhance early exploration, we implement a dynamically decaying ϵ with a decay rate λ. This method enables the agent to gather crucial environment knowledge in the early stages of training and gradually shift towards exploiting this knowledge to maximize rewards.

In Fig. 6.2, the proposed DRL algorithm is depicted. Initially, the DQN processes the current state of the environment s_t and selects an action a_t based on the current policy π. The reward r_t is then calculated, and the next state s_{t+1} is determined. The experience tuple (s_t, a_t, r_t, s_{t+1}) is stored in the replay memory $\mathcal{M}$ until the memory reaches its full capacity. Once the memory is full, the DQN begins training by randomly sampling a mini-batch X of experiences from the replay memory $\mathcal{M}$ to update its parameters θ using (6.19). After every κ episodes, the parameters θ' of the Target Q-network are updated by copying the parameters θ from the DQN. The training continues until the episodes are completed.

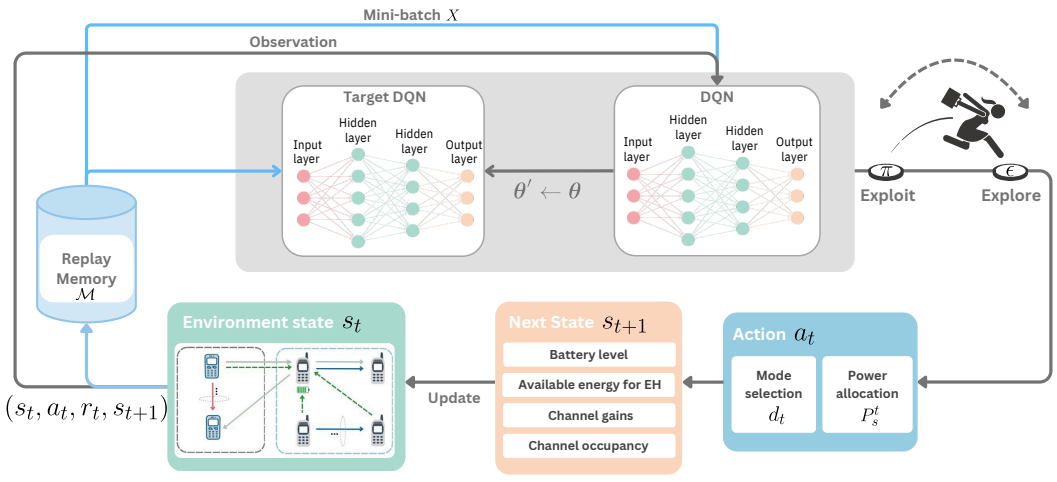


Figure 6.2: The proposed ϵ-greedy-based DQN algorithm.

The algorithm for joint power control and channel access coordination, based on the CIoT system operation and the proposed DRL framework, is provided in Algorithm 5. It is important to note that the use of non-linear function approximators, such as other DRL methods, eliminates any guarantees of convergence [96, 97]. Consequently, it is very challenging to provide a precise upper bound or confirm convergence [84]. However, the simulation results demonstrate consistent learning without requiring modifications or additional assumptions about the environment.

6.4.3 Simulation Results

This section presents the results of comprehensive simulations showing the performance of the proposed DRL strategy for joint power control and channel access coordination in WPT-enabled CIoT networks.

Setup

Table 6.1 contains the simulation parameters used. The channel power gains g_{ss}^t and g_{sp}^t follow an exponential distribution with a mean of 0.1 and 0.2 respectively. Without loss of generality, we consider $g_{sp}^t = g_{ps}^t$. For our proposed DQN architecture, we utilize four fully connected layers, each with a specific number of neurons: $j = 7$, $h_1 = 128$, $h_2 = 64$, and $z = 22$. The Leaky ReLU's hyperparameter is set to $\alpha = 0.02$. Regarding the learning rate, we begin with $\eta = 4 \times 10^{-4}$, which then decreases by a factor of 50% every 500 episodes. An advanced SGD-based parameter update method called Adaptive Moment Estimation (Adam) is used during training, as it offers faster computation time. A penalty value of $\phi = 7$ is assigned when the CIoT agent violates the constraints outlined in (6.11) during the training process.

Algorithm 5 Algorithm for joint power control and channel access coordination to solve problem (6.11).

Input: Cognitive radio environment simulator and its parameters.
Output: Optimal action a_t in each time slot t.
Initialize experience replay memory $\mathcal{M}$ with size m.
Initialize battery level B_0.
Initialize $\forall \theta \in \boldsymbol{\theta}, \quad \theta \sim \mathcal{N}(0, \frac{2}{v_i})$, and set $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}$.
Initialize η and set the scheduler's reduction factor and patience period.
Initialize ϵ and set the decay rate λ.
Initialize γ and κ.
for episode = 1 to episodes **do**
    for $t = 1$ to T **do**
        Observe the state s_t.
        if $\mathcal{M}$ is not full **then**
            Sample a random action a_t.
            Get the reward r_t and observe the next state s_{t+1}.
            Store (s_t, a_t, r_t, s_{t+1}) in $\mathcal{M}$.
        else
            Sample $p \sim \mathcal{U}(0, 1)$.
            if $p > \epsilon$ **then**
                Get action a_t according to current policy π.
            else
                Sample a random action a_t.
            end if
            Get the reward r_t.
            Sample a mini-batch X from $\mathcal{M}$.
            Predict Target Q-values using:

$$\mathbf{r}_t + \gamma \max_{\mathbf{a} \in \boldsymbol{\mathcal{A}}} Q^{\pi}(\mathbf{s}_{t+1}, \mathbf{a}; \boldsymbol{\theta}')$$

            Predict Q-values using $Q^{\pi}(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\theta})$.
            Calculate the loss in (6.17).
            Update $\boldsymbol{\theta}$ of DQN online using (6.19).
            if (episode $\cdot t$) $\bmod \kappa = 0$ **then**
                Update $\boldsymbol{\theta}'$ of Target DQN: $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}$.
            end if
        end if
        Update η using scheduler.
        Update ϵ.
        Update the state $s_{t+1} \leftarrow s_t$.
    end for
end for

Table 6.1: Simulation parameters for the proposed DRL-driven WPT-enabled CIoT Network.

Parameters	**Value**
Number of time slots T	30
Duration of each time slot τ	1 s
Number of PU transmission slots L	20
Transmission power of PU P_p	0.2 W
Interference threshold I_{th}	0.01 W
Battery capacity B_{max}	0.5 W
Transmission power range of CIoT Tx P	0.01 $\sim$ 0.1 W
Number of competing CIoT devices N	[2, 10]
Transmission power of CIoT devices P_n	0.1 W
Noise power σ^2	1e-3 W
Experience replay memory size m	10,000
Training episodes	2500
Mini-batch size	100
Learning rate η	$4 * 10^{-4}$
Learning rate reduction factor	50%
Learning rate patience period	500 episodes
Penalization ϕ	7
Discount factor γ	0.99
Exploitation rate ϵ	0.1
Exploration decay rate λ	$1 * 10^{-8}$
Leakiness parameter α	0.02
Update rate of Target DQN κ	200

To evaluate the effectiveness of our proposed DRL strategy, which employs the DQN-driven design discussed in Section 6.4.2, we conduct a comparative analysis with the following strategies:

- A learning strategy in which the CIoT agent selects an action a_t at each step of the environment based on the policy derived from the proposed DQN outlined in [98].
- A random strategy, where the action a_t is chosen randomly at each step from the action space, without any intelligent decision-making or cognition.
- A fixed strategy, based on rule-based approaches derived from the constraints outlined in (6.11), which does not incorporate any learning processes. In this approach, the CIoT agent determines its action a_t at each step according to these predefined rules.

In our study, we analyze both the sum rate and the achievable reward over the course of the training episodes. To evaluate these metrics, we apply a weighted moving average to reduce the effect of short-term fluctuations, with the goal of identifying trends in the sum rate and reward. This method balances recent changes with historical data, providing a more comprehensive analysis of the training

episodes. The weighted moving average is calculated as follows

$$\text{average}_{\text{new}} = (1 - \delta) \times \text{average}_{\text{old}} + \delta \times \text{value}, \tag{6.20}$$

where δ is the weight assigned to the new value and $1 - \delta$ is the weight assigned to the previous average. For our considerations, we set δ to 0.01.

Results and Analysis

In Fig. 6.3, we display the Average Sum Rate (ASR) attained by the CIoT agent employing our proposed DQN-based strategy. During the initial stages of training, the CIoT agent utilizing our approach incurs penalties due to exploratory actions, resulting in lower performance compared to the fixed strategy and performance similar to the random strategy. However, as training progresses, the CIoT agent using our strategy gradually surpasses both the fixed and random strategies. This improvement is attributed to the agent's ability to consider long-term performance, leading to more efficient resource utilization and ASR. It is important to note that the fixed strategy does not incur penalties, as it follows predefined operational rules and constraints. However, despite avoiding

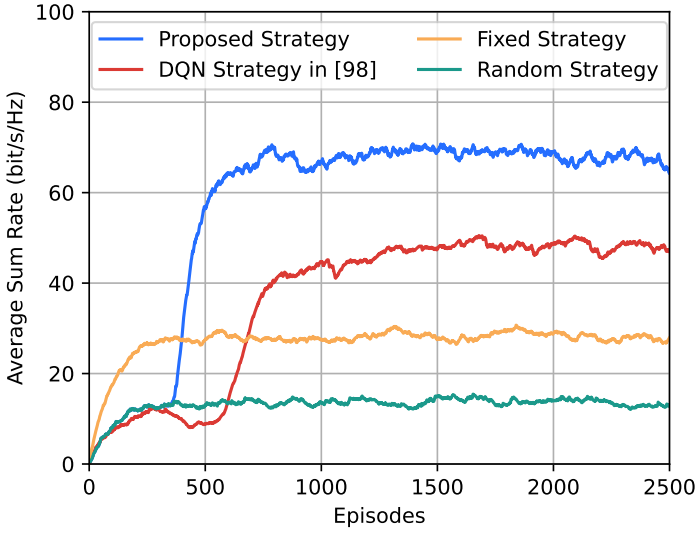


Figure 6.3: The CIoT agent's ASR performance across training episodes, employing diverse strategies for joint power control and channel access coordination.

penalties, the fixed strategy does not achieve maximum throughput. Overall, the performance gap between our proposed learning approach and the fixed and random strategies highlights the limitations of relying solely on predefined rules or random actions to optimize data rates in dynamic CIoT environments. As a result, Fig. 6.3 emphasizes the importance of utilizing learning algorithms to effectively optimize data rates in response to the continuously changing radio environment. Furthermore, we observe that both our proposed DRL strategy and the method used in [98] converge, validating the effectiveness of our training approach. However, a noticeable performance gap emerges between our proposed DQN and the approach in [98], highlighting not only the faster convergence and adaptability of our DQN, but also its consistent ability to deliver superior performance.

In Fig. 6.4, the average reward achieved by the CIoT agent in each training episode is shown. As seen in the figure, our proposed method consistently outperforms all other strategies in terms of average reward. Comparing Fig. 6.3 and Fig. 6.4, it is evident that the ASR is consistently higher than the average reward for all strategies. This discrepancy arises because the average reward plot takes into account both negative rewards (penalties) and positive rewards (sum rate), whereas the ASR plot only reflects positive rewards. Fig. 6.4 demonstrates that both our proposed DRL strategy and the method in [98] show convergence. However, the CIoT Tx using the DQN from [98] incurs substantial penalties early in training, ultimately converging to a lower average reward compared to our approach, which leads to slower convergence. In comparison to the fixed strategy, the approach

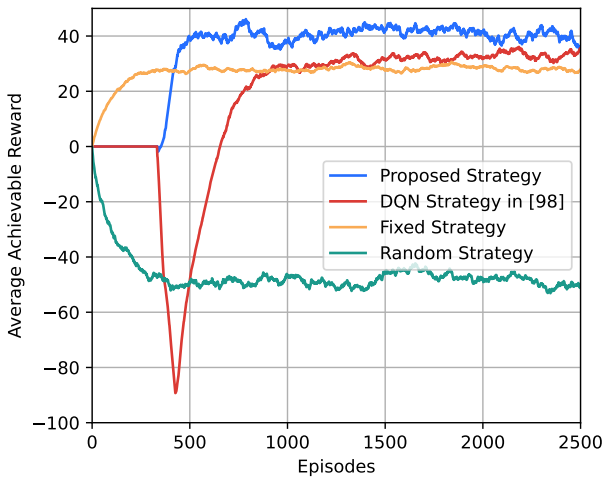


Figure 6.4: The CIoT agent's average achievable reward during episodes of training while utilizing various types of strategies for joint power control and channel access coordination.

in [98] initially yields almost identical average rewards. However, upon examining Fig. 6.3, it becomes clear that the approach in [98] achieves a considerably higher ASR than the fixed strategy. This difference occurs because the CIoT Tx using the DQN in [98] attempts to learn a policy that maximizes the ASR, leading to penalties during training, whereas the fixed strategy avoids penalties. As shown in Fig. 6.4, the random strategy employed by the CIoT Tx leads to numerous penalties over time, resulting in an overall negative reward.

Fig. 6.5 shows the effect of different ϵ values used in the ϵ-greedy exploration strategy on the average reward of the CIoT agent employing our DRL method. As the values of ϵ vary, changes in the CIoT agent's behavior and the corresponding effect on the average reward are observed. Specifically, when $\epsilon = 0.1$, the highest average reward is achieved. This indicates that lower values of ϵ provide an optimal balance between exploration and exploitation, enabling the CIoT agent to make decisions based on its acquired knowledge while occasionally exploring new possibilities. However, as ϵ values increase, a decline in the average reward is seen. The poorest performance is observed at $\epsilon = 0.7$, suggesting that the CIoT agent is overly focused on exploration. This excessive exploration results in frequent penalties, as the CIoT agent selects actions randomly or with minimal consideration of its learned knowledge, leading to suboptimal choices and a decrease in performance.

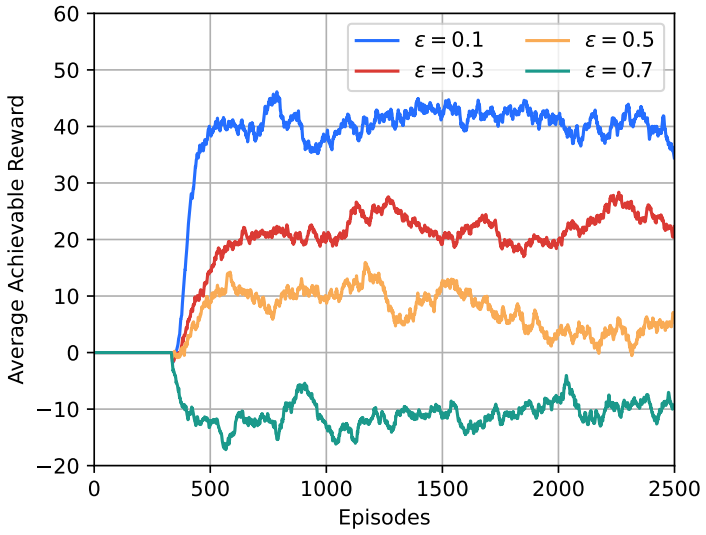


Figure 6.5: Effect of the ϵ parameter in the ϵ-greedy exploration strategy on the average achievable reward of the CIoT agent using our proposed DRL-driven strategy.

Fig. 6.6 illustrates the impact of the initial battery level B_0 on the ASR achieved by the CIoT agent using our proposed DRL approach. The lowest performance is observed when the CIoT agent starts with $B_0 = 0$, leading to a decline in performance as the agent frequently incurs penalties for attempting data transmission without adequate energy. In this case, the CIoT agent focuses on energy harvesting to ensure enough energy for transmission whenever possible. On the other hand, when the initial battery level is non-zero ($B_0 > 0$), the agent becomes less focused on energy harvesting and can concentrate more on identifying actions that maximize data transmission. Consequently, the optimal scenario is when the CIoT agent begins training with a fully charged battery ($B_0 = B_{max}$), allowing the agent to prioritize actions that maximize its reward while also contributing to a longer network lifetime.

Fig. 6.7 illustrates the effect of increasing the maximum battery capacity B_{max} on the ASR of the CIoT agent. As shown, our proposed approach consistently outperforms other strategies across a range of B_{max} values. This demonstrates the flexibility of our approach in optimizing the ASR, whether the battery capacity is large or small. With a higher B_{max}, energy harvesting improves, enabling the CIoT agent to sustain data transmission over extended periods, which results in higher ASR values. Moreover, increasing B_{max} reduces battery overflow (harvesting energy beyond B_{max}),

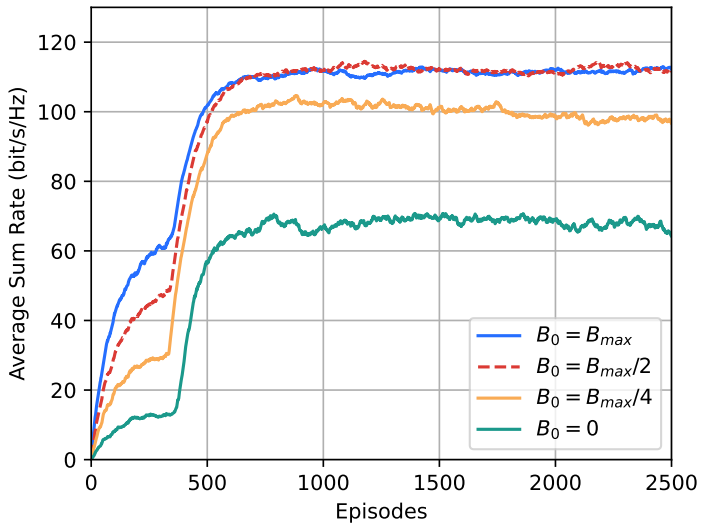


Figure 6.6: The effect of starting battery level B_0 on the ASR of the CIoT agent using our proposed DRL-driven approach.

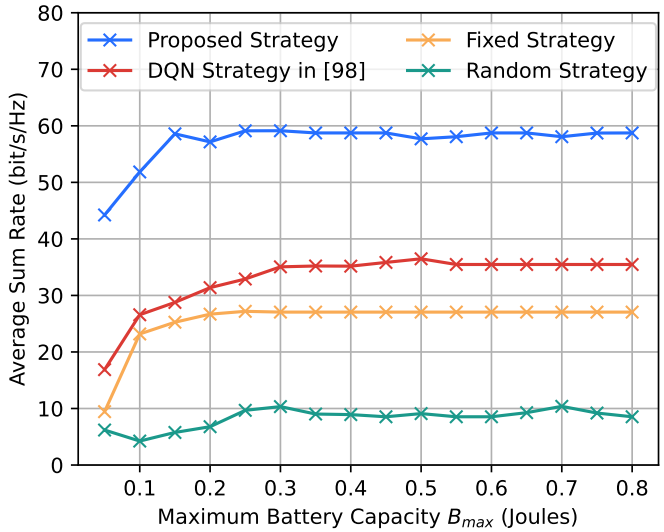


Figure 6.7: The effect of the maximum battery capacity B_{max} of the CIoT agent on the ASR while using different types of strategies for joint power control and channel access coordination.

leading to fewer penalties for the SU. However, after a certain threshold, further increases in battery capacity yield diminishing returns, indicating a point of resource saturation where additional capacity does not result in proportional improvements in performance.

Fig. 6.8 shows the impact of varying the total number of time slots T on the ASR of the CIoT Tx. As T increases, there is a corresponding rise in ASR for all strategies. This trend is due to the larger number of time slots, which provide the CIoT Tx with more opportunities for transmission, resulting in a higher ASR. However, our proposed DRL strategy consistently outperforms all other strategies in terms of ASR across all values of T. In situations with fewer time slots (small T values), the CIoT system faces limitations from the reduced number of transmissions, leading to lower ASR. Interestingly, for smaller T values, both the fixed strategy and the approach in [98] demonstrate similar performance, suggesting that the latter is less effective at maximizing ASR in such scenarios. In contrast, the performance of our proposed DRL strategy significantly exceeds that of all other strategies, highlighting its exceptional effectiveness and adaptability across different communication system conditions.

To demonstrate the adaptability of our proposed approach to various primary network scenarios, we vary the number of time slots occupied by the PU and analyze its impact on the performance of the

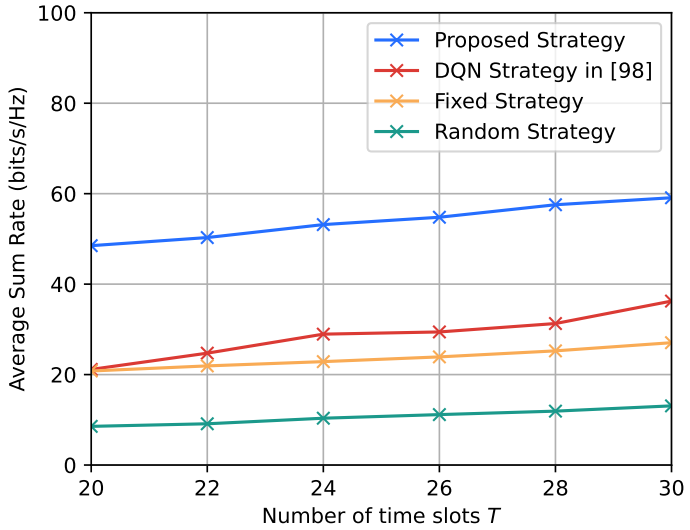


Figure 6.8: The effect of the number of time slots T on the CIoT agent's ASR under different strategies.

CIoT agent. Fig. 6.9 shows the effect of the PU's occupied slots L on the ASR of the CIoT agent, with the total number of time slots fixed at $T = 30$. Notably, our DRL strategy consistently outperforms all other strategies for different values of L. As L increases, the CIoT Tx gains more opportunities for energy harvesting. However, to adhere to the interference threshold I_{th}, the transmission power is reduced during the L slots, leading to a more cautious approach to data transmissions and a decline in ASR. It is important to note that for higher values of L, particularly when $L \geq 28$, all strategies attain similar performance. In this case, the PU occupies nearly all available slots, leaving the CIoT agent with very limited action choices, resembling a fixed strategy.

Fig. 6.10 evaluates the achievable ASR of the CIoT Tx using our proposed DRL approach across various PU transmission power levels P_p, comparing it to benchmark strategies. The figure clearly shows that as P_p increases, the ASR improves for all strategies, reflecting enhanced energy harvesting with higher PU transmission power. Notably, our proposed DRL strategy surpasses all other strategies by a significant margin. This highlights our approach's ability to optimally leverage harvested energy, maximizing the ASR while complying with operational constraints. Even in scenarios with low P_p, our DRL approach demonstrates superior learning in power control and channel access strategies, resulting in better ASR than alternative strategies. In contrast, the DQN in [98] struggles to effectively exploit the energy gains from higher PU power, indicating suboptimal

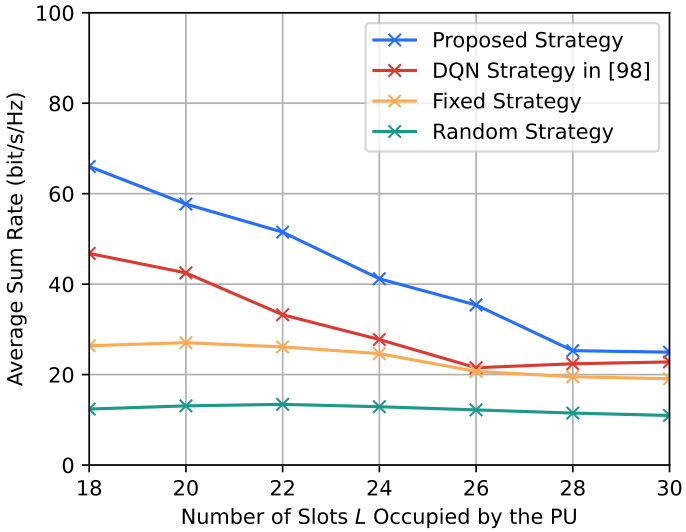


Figure 6.9: The effect of the number of PU transmission slots L on the CIoT agent's ASR under different strategies.

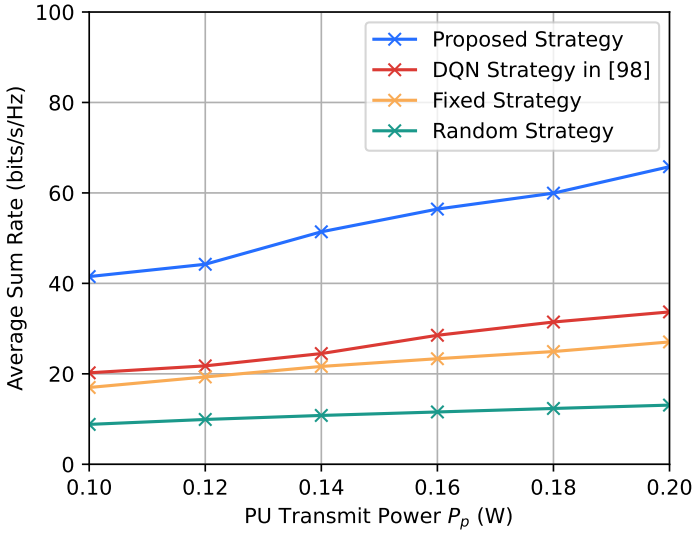


Figure 6.10: The effect of the PU transmit power P_p on the CIoT agent's ASR under various strategies.

performance. Although it also employs a DQN-based approach, it fails to fully capitalize on harvested energy, further emphasizing the effectiveness of our proposed approach.

In Fig. 6.11, we increase the number of competing CIoT devices from $N = 2$ to $N = 10$ and examine the CIoT agent's interference rate throughout the training episodes. The interference rate

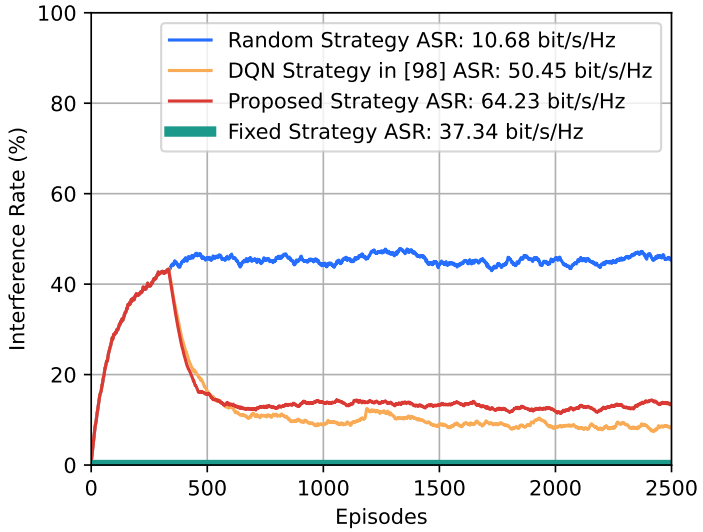


Figure 6.11: The CIoT agent's interference rate under various strategies when the number of competing CIoT devices N= 10. The legend shows the ASR at convergence.

is defined as the percentage of time slots during which the CIoT agent interferes with other CIoT devices in the network. The legend of the figure also shows the ASR at convergence. Notably, the random strategy results in the highest interference. In contrast, the fixed strategy ensures no interference among users, as it adheres to the CR constraint, which prevents transmission when other CIoT devices are present, resulting in a 0% interference rate. For both our proposed DRL approach and the DQN in [98], interference initially increases during the exploration phase. However, as training progresses, the interference decreases and stabilizes. While our proposed DRL approach results in slightly higher interference compared to the DQN in [98], it achieves a higher ASR, indicating that our approach ultimately provides a more effective joint power control and channel access coordination strategy. Furthermore, it is clear that as the number of CIoT devices N increases, there is a slight decline in performance when comparing the ASR values at convergence in Fig. 6.3 with those shown in Fig. 6.11. This performance drop is due to the increased competition for available time slots, resulting in fewer transmission opportunities for the CIoT agent.

6.5 Joint Time and Power Management in SWIPT-EH CIoT

Unlike Wireless Power Transfer (WPT), Simultaneous Wireless Information and Power Transfer (SWIPT) allows devices to divide a communication slot between energy harvesting and data transmission using the Time Switching (TS) protocol, rather than limit the slot to a single function. Therefore, dynamic decision-making is essential for energy-constrained CIoT devices to intelligently select the optimal TS ratio and transmission power. In this section, we design a DRL algorithm that enables devices to develop *self-adaptation* capabilities and optimize their operations in a SWIPT-enabled CIoT network, aiming to maximize the long-term achievable sum rate.

6.5.1 System Model and Problem Formulation

Cognitive IoT System

Consider a CIoT network comprising a Tx-Rx pair that operates alongside a primary Tx-Rx pair, as illustrated in Fig. 6.12. The communication system is structured into time slots, each with a duration of τ seconds, and a total of T slots of equal length. The CIoT Tx is equipped with a finite battery capacity of B_{max} and supports SWIPT-EH. To regulate the SWIPT process within each time slot t, the Time Switching (TS) protocol is utilized, governed by the TS factor $0 \leq \rho_t \leq 1$. Specifically, the CIoT agent can switch between EH and data transmission, where each time slot τ is divided into two segments: $\rho_t \tau$ is allocated for EH, while the remaining $(1-\rho_t)\tau$ is used for data

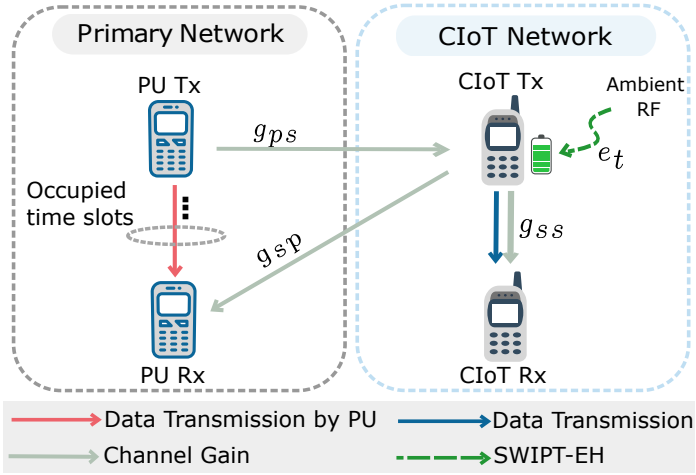


Figure 6.12: Illustration of the designed system model for the SWIPT-enabled CIoT network under study.

transmission.

The PU Tx is allocated L time slots for transmission, during which it operates at a constant transmit power of P_p^t. The activity of the PU Tx in slot t is indicated by the status variable ω_p^t, which is set to 1 when active and 0 otherwise. In underlay CR, the CIoT Tx is allowed to transmit in the same slot as the PU Tx, provided it adheres to the interference threshold I_{th}. This constraint is expressed as $P_s^t g_{sp}^t \leq I_{th}$, where P_s^t represents the transmit power of the CIoT Tx, and g_{sp}^t denotes the channel power gain between the CIoT Tx and the PU Tx. The channel power gains g_{ss}^t, g_{ps}^t, and g_{sp}^t are modeled as independently and identically distributed (i.i.d.) Rayleigh fading channels, remaining constant within each time slot [14]. The channel power gain g_{ij}^t follows an exponential distribution with the PDF given by $f_{g_{ij}^t}(y) = \lambda_{ij} \exp(\lambda_{ij} y)$. The fading parameter λ_{ij} is determined by the device separation distance d_{ij} and the path loss exponent α, expressed as $\lambda_{ij} = d_{ij}^{-\alpha}$.

When the channel is unoccupied, the achievable rate of the CIoT Tx during the t-th time slot is

$$R_0^t = \log_2\left(1 + \frac{P_s^t g_{ss}^t}{\sigma^2}\right), \tag{6.21}$$

where σ^2 denotes the variance of the channel noise. If the PU Tx occupies the channel during the t-th time slot, the achievable rate of the CIoT Tx is reduced due to interference from the PU, as specified by

$$R_1^t = \log_2\left(1 + \frac{P_s^t g_{ss}^t}{P_p^t g_{ps}^t + \sigma^2}\right). \tag{6.22}$$

The EH process is modeled as an energy arrival system with independently and identically distributed time slots. At $t = 0$, the harvested energy is initialized as $e_0 = 0$. The energy from ambient sources follows a Gamma distribution, denoted as $\hat{e} \sim \Gamma(k, \beta)$, where k and β represent the shape and scale parameters, respectively. As a result, the harvested energy in each time slot is given by $e_t = \mu \hat{e}$, where $0 \leq \mu \leq 1$ represents the energy conversion efficiency. In the subsequent time slot $t+1$, the available battery level is updated based on the CIoT device's chosen parameters, (ρ_t, P_s^t), as follows

$$B_{t+1} = \min\left\{B_t + \rho_t e_t \tau - (1-\rho_t) P_s^t \tau, B_{max}\right\}. \tag{6.23}$$

ρ_t indicates the fraction of the time slot's duration that the CIoT device decided to harvest and (1-ρ_t)

indicates the remaining time slot's duration that the CIoT will transmit data.

Optimization Problem Formulation

In the studied network, the CIoT Tx aims to optimize both its TS factor ρ_t and transmit power P_s^t to maximize its long-term achievable sum rate while accounting for the interference threshold I_{th}, the available battery energy B_t, and the energy harvested e_t. Thus, the maximization of the CIoT Tx's rate can be formulated as a constrained optimization problem as follows

$$\max_{\rho_t, P_s^t} \sum_{t=1}^{T} (1-\rho_t)\tau\left[(1-\omega_p^t)R_0^t + \omega_p^t R_1^t\right] \tag{6.24a}$$

$$\text{s.t.} \sum_{t=1}^{T} P_s^t(1-\rho_t)\tau \leq B_0 + \sum_{t=0}^{T-1} e_t\tau, \ \ \forall T \tag{6.24b}$$

$$0 \leq P_s^t(1-\rho_t)\tau \leq B_t, \ \ \rho_t \in [0,1] \tag{6.24c}$$

$$\omega_p^t g_{sp}^t P_s^t \leq I_{th}, \ \ \omega_p^t \in \Omega \triangleq \{0,1\}. \tag{6.24d}$$

We address the optimization problem by modeling the action-taking process of the CIoT Tx as a stochastic control process within a discrete-time system. According to the Markov property, the next system state depends solely on the current state and action. In our framework, the stochastic nature of state variables, such as PU activities and energy arrivals, adheres to this property. Consequently, determining the optimal operational strategy can be framed as a Markov Decision Process (MDP). This MDP is defined by the tuple $(\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{P}}, \boldsymbol{\mathcal{R}}, T)$, where $\boldsymbol{\mathcal{S}}$ is the set of environment states, $\boldsymbol{\mathcal{A}}$ is the set of possible actions, $\boldsymbol{\mathcal{P}}$ is the state transition probabilities, $\boldsymbol{\mathcal{R}}$ specifies the rewards associated with each state-action pair, and T indicates the time step.

In practical scenarios, accurately determining the PDF of energy arrivals and channel fading is challenging. Additionally, the CIoT network lacks precise knowledge of the state transition probabilities governing the primary network's occupancy states, making it infeasible to determine $\boldsymbol{\mathcal{P}}$ exactly. To address this, we adopt a model-free MDP approach and design a Deep Reinforcement Learning (DRL) framework to approximate $\boldsymbol{\mathcal{R}}$ based on $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{A}}$, eliminating the need for $\boldsymbol{\mathcal{P}}$. Consequently, the model-free MDP formulation is reduced to the tuple $(\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{R}}, T)$, where the

components are defined as follows

(1) **State Space $\mathcal{S}$:** The state space comprises all potential states over T time slots. At each state s_t, the CIoT agent must account for multiple factors, including the current battery level B_t, the energy harvested in the previous time slot e_{t-1}, the occupancy status of the slot by the PU Tx, and the channel power gains g_{ps}^t, g_{sp}^t, and g_{ss}^t. Therefore, the environment state at any given time slot t is represented as $s_t = \{B_t, e_{t-1}, \omega_p^t, g_{ps}^t, g_{sp}^t, g_{ss}^t\}$.

(2) **Action Space $\mathcal{A}$:** The action space comprises all feasible actions the CIoT agent can take. Given the current environment state s_t, the agent must determine the TS factor ρ_t and the appropriate transmit power P_s^t. Consequently, the action selected by the CIoT agent at each time slot t is represented as $a_t = [\rho_t, P_s^t]$, where $\rho_t \in [0, 1]$ and $P_s^t \in P$.

(3) **Reward $\mathcal{R}$:** The reward is defined based on the achievable rate while ensuring compliance with all constraints outlined in (6.24). If the CIoT agent selects an action a_t that violates any of these constraints, it incurs a negative reward (penalty). Accordingly, the reward r_t for the CIoT at each time slot t is given by:

$$r_t = \begin{cases} \rho_t' R_0^t & \omega_p^t = 0, 0 \leq P_s^t \rho_t' \tau \leq B_t \\ \rho_t' R_1^t & \omega_p^t = 1, 0 \leq P_s^t \rho_t' \tau \leq B_t, P_s^t g_{sp}^t \leq I_{th} \\ -\phi & \text{others,} \end{cases} \tag{6.25}$$

where ρ_t' represents $(1 - \rho_t)$.

(4) **Time Step T:** We characterize each progression from time slot t to $t + 1$ as a single step and systematically evaluate each state-action across all time slots T.

6.5.2 Double Deep Q-Network with Upper Confidence Bound Exploration Strategy

We present our proposed DRL framework, which enables the agent to effectively approximate the state-value function and derive a policy π for selecting actions based on the current state of the environment. The objective of π is to maximize the long-term cumulative reward (rate) while

ensuring compliance with the constraints of the CIoT system. Given that energy-constrained CIoT devices operate at low transmission power levels and the Time Switching (TS) ratio is restricted to $0 \leq \rho \leq 1$, the action space remains relatively small. This characteristic allows for efficient discretization, thereby reducing computational complexity.

The Proposed Double Deep Q-Network Architecture

Deep Q-Networks (DQNs) have been widely employed to determine optimal actions in discrete action spaces. However, they are prone to overestimation bias, which can lead the learning agent to favor overly optimistic actions, ultimately resulting in suboptimal performance. To mitigate this issue, we adopt a Double Deep Q-Network (DDQN) architecture to estimate the cumulative reward (Q-value) for a given action a in state s. The DDQN is designed to optimize its parameters $\boldsymbol{\theta}$ such that $Q^{\pi}(s, a; \boldsymbol{\theta}) \approx Q^{\pi}(s, a)$. It is implemented as a lightweight fully connected neural network, where the input layer comprises j neurons corresponding to the dimensions of the state space $\boldsymbol{\mathcal{S}}$. The network further consists of two hidden layers with h_1 and h_2 neurons, respectively, and an output layer containing z neurons. The DDQN parameters, $\boldsymbol{\theta} = \{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}$ for each network layer i, represent the associated weights and biases.

During training, a Target DDQN is utilized, which initially mirrors the DDQN. As training advances, the parameters of the Target DDQN are updated at a slower rate compared to those of the DDQN, often across several training steps. The proposed DDQN is trained using the Mean Squared Error (MSE) loss $\mathcal{L}$ to compute the MSE between predicted and target Q-values for a mini-batch of state-action pairs $(\mathbf{s}, \mathbf{a})$ as

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}\left[\left[\mathbf{r}_t + \gamma Q^{\pi}\left(\mathbf{s}_{t+1}, \arg\max_{\mathbf{a} \in \boldsymbol{A}} Q^{\pi}(\mathbf{s}_{t+1}, \mathbf{a}; \boldsymbol{\theta}); \boldsymbol{\theta}'\right) - Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t; \boldsymbol{\theta})\right]^2\right], \tag{6.26}$$

The Target DDQN's parameter set $\boldsymbol{\theta}'$ is updated using an experience replay buffer, which stores past experiences (s_t, a_t, r_t, s_{t+1}) to reduce temporal correlations. Once the buffer reaches a predefined threshold κ, mini-batches of experiences are randomly sampled to update the DDQN parameters.

During training, the goal is to minimize the loss function $\mathcal{L}(\boldsymbol{\theta})$ over a mini-batch of state-action

pairs, with the objective $\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathbf{s}, \mathbf{a})$. The backpropagation algorithm is used to compute the gradient $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathbf{s}, \mathbf{a})$, which quantifies how the loss changes with respect to the parameters of the DDQN for the given state-action pairs. Using Stochastic Gradient Descent (SGD) and the calculated gradients, the DDQN parameters are updated as $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathbf{s}, \mathbf{a})$. The learning rate $0 < \eta < 1$ is fine-tuned with the Adaptive Moment Estimation (Adam) optimizer for more efficient computation, while a learning rate scheduler is employed to gradually reduce the learning rate, ensuring stable learning and faster convergence.

Upper Confidence Bound Exploration Strategy

To allow the CIoT agent to explore the environment, discover optimal strategies, and balance the exploration-exploitation trade-off, we employ the Upper Confidence Bound (UCB) algorithm. This algorithm updates the Q-values with $\overline{Q}^{\pi}(s, a) = Q^{\pi}(s, a) + U_a^t$, where U_a^t represents the computed expected reward, defined as

$$U_a^t = \hat{r}_a^t + \sqrt{\frac{c' \ln t}{C_a^t}}, \tag{6.27}$$

with c' being a hyperparameter of the UCB algorithm. The computed expected reward U_a^t incorporates the estimated reward $\hat{r}_a^t$ along with an adjustment factor that depends on the current time period (frame number * T + t) and the number of times action a has been chosen, denoted C_a^t. If action a_t has been selected C_a^t times by the end of time slot t (ranging from 0 to t), then $\hat{r}_a^t$ is computed as

$$\hat{r}_a^t = \frac{\sum_{i=1}^{C_a^t} r_{a,i}^t}{C_a^t}, \tag{6.28}$$

where $r_{a,i}^t$ represents the reward for action a_t during the ith selection. Following this, the action selected by the UCB algorithm is integrated into the DDQN training, and both the action count C_a^t and the expected reward $\hat{r}_a^t$ are updated accordingly.

The update of the Q-value using the UCB strategy is depicted in Fig. 6.13. The training procedure for the DRL algorithm is detailed in Algorithm 6. The robustness of the UCB algorithm stems from the assumption that the agent receives immediate feedback to update its confidence. However, it is

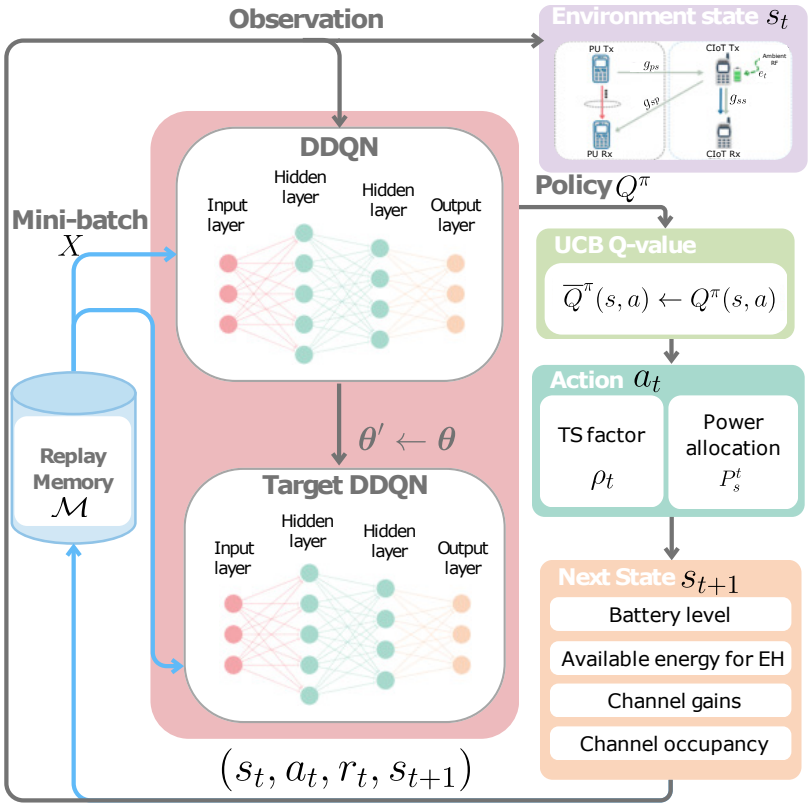


Figure 6.13: The proposed double deep Q-network architecture with upper confidence bound exploration strategy.

important to note that in real-world scenarios, delays can cause feedback from different time steps to be revealed simultaneously, meaning the CIoT agent might receive multiple pieces of feedback at once or receive none at certain times.

Algorithm 6 The proposed UCB-driven DRL algorithm to solve (6.24).

Input: Cognitive IoT environment simulator and parameters.
Output: Optimal action a_t in each time slot t.
Initialize experience replay memory $\mathcal{M}$ with size m.
Initialize B_0, η, γ, κ, and c'.
for episode = 1 to episodes **do**
  for $t = 1$ to T **do**
    Observe the state s_t.
    if $\mathcal{M}$ is not full **then**
      Sample a random action a_t.
    else
      Calculate $U_a^t \leftarrow \hat{r}_a^t + \sqrt{\frac{c' \ln t}{C_a^t}}$.
      Adjust Q-value: $\overline{Q}^{\pi}(s,a) \leftarrow Q^{\pi}(s,a) + U_a^t$.
      Get action a_t according to the policy based on the adjusted Q-value.
    end if
    Get the reward r_t using (6.25).
    Observe the next state s_{t+1}.
    Store (s_t, a_t, r_t, s_{t+1}) in $\mathcal{M}$.
    Update action count: $C_a^t \leftarrow C_a^t + 1$.
    Update $\hat{r}_a^t \leftarrow \frac{\sum_{i=1}^{C_a^t} r_{a,i}^t}{C_a^t}$.
    Sample a mini-batch X from $\mathcal{M}$.
    Predict Target Q-values using:

$$\mathbf{r}_t + \gamma \max_{\mathbf{a} \in \boldsymbol{\mathcal{A}}} Q^{\pi}(\mathbf{s}_{t+1}, \mathbf{a}; \boldsymbol{\theta}')$$

    Predict Q-values using $Q^{\pi}(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\theta})$.
    Calculate the loss in (6.26).
    Update $\boldsymbol{\theta}$ of DDQN online.
    if $(episode \cdot t) \bmod \kappa = 0$ **then**
      Update $\boldsymbol{\theta}'$ of the target DDQN: $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}$.
    end if
  end for
  Update ϵ and update the state $s_{t+1} \leftarrow s_t$.
  Update η using the scheduler.
end for

6.5.3 Simulation Results

This section provides an in-depth analysis of simulation results, demonstrating the effectiveness of the proposed DRL strategy for optimizing both time and power management in SWIPT-enabled CIoT networks.

Setup

We consider a channel noise variance of $\sigma^2 = 10^{-3}$ and a path loss exponent of $\alpha = 4$. The device distances are $d_{sp} = d_{ps} = 1.8\,\text{m}$ and $d_{ss} = 1.5\,\text{m}$. Time-slotted transmissions are considered over $T = 30$ slots (each lasting $\tau = 1\,\text{s}$), with the PU Tx using $L = 18$ slots at $P_p = 0.2\,\text{W}$ and an interference threshold of $I_{th} = 0.1\,\text{W}$. The CIoT Tx's battery capacity is $B_{max} = 0.5\,\text{W}$, with dynamic selection of $\rho_t \in [0, 0.1, \ldots, 1]$ and $P_s^t \in [0, 0.01, \ldots, 0.1]$. The harvested energy at a time slot t follows $e_t \sim \Gamma(0.5, 1)$ and the energy efficiency factor is $\mu = 0.9$. The DDQN architecture has four layers with $j = 6$, $h_1 = 512$, $h_2 = 128$, and $z = 121$ neurons, updating the target network every 200 iterations. Training uses an initial learning rate of 2×10^{-4}, halved every 500 episodes, over 2500 episodes with mini-batches of 80 frames. Constraint violations incur a penalty $\phi = 7$. The Target DDQN is updated every 200 iterations. The replay buffer holds $\kappa = 333$ experiences, with a discount factor $\gamma = 0.99$ and UCB hyperparameter $c' = 2.5$.

Results and Analysis

In Fig. 6.14, we present a thorough comparison of the Average Sum Rate (ASR) achieved by the CIoT agent using our proposed DRL approach against various existing benchmarks. The strategies included for comparison are as follows: the random strategy, where actions are selected randomly at each time step; the DRL strategy from [87], which utilizes a fixed TS factor ($\rho_t = 0.5$) and a learnable transmission power P_s^t; and the DQN and Dueling Deep Q-Network (D3QN) strategies from [99]. We employ both the ϵ-greedy and UCB methods to balance the exploration-exploitation trade-off for each of these learning strategies.

At the beginning of the training process, all DRL strategies focus on action exploration, which results in penalties, as depicted in Fig. 6.14. However, as training progresses, our DDQN-UCB

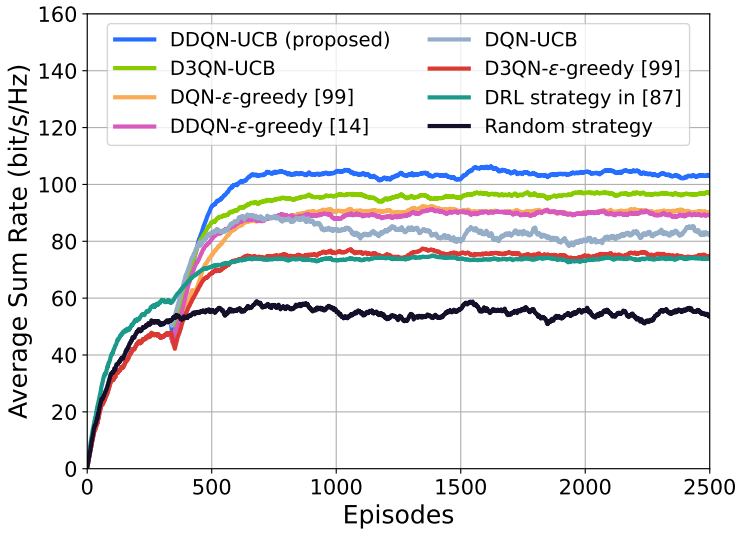


Figure 6.14: Benchmarking the ASR performance of our proposed DDQN-UCB strategy in comparison to the existing strategies in the literature.

strategy gradually outperforms all the other methods. This improvement is primarily due to the proposed UCB exploration, which effectively explores actions that optimize long-term performance, thus improving resource allocation efficiency and increasing ASR. Nonetheless, this enhancement is not solely attributed to UCB; replacing the DRL framework with DQN or Dueling Deep Q-Network (D3QN) while using UCB does not lead to the optimal strategy, highlighting that the combination of the DDQN architecture and UCB is superior. Additionally, the performance gap between DDQN-UCB and D3QN-UCB clearly demonstrates that more complex architectures, such as D3QN, do not always guarantee improved performance, which can be attributed to the extra layers in D3QN. Moreover, restricting the TS factor ρ as in [87] and focusing exclusively on power optimization does not lead to the most effective strategy for maximizing performance. Neglecting to jointly optimize both the TS factor and transmission power results in suboptimal outcomes.

In Fig. 6.15, we examine the effect of the number of slots occupied by the PU, denoted by L, on the ASR of the CIoT agent using our proposed DDQN-UCB strategy. The results are compared across different numbers of time slots T. As shown in the figure, an increase in L leads to a decrease in ASR, primarily due to the additional constraints on the CIoT agent's actions. With a higher number of PUs, the CIoT agent must adhere to the interference threshold across more slots, which increases the likelihood of penalties and reduces the transmission data rate. On the other hand, when

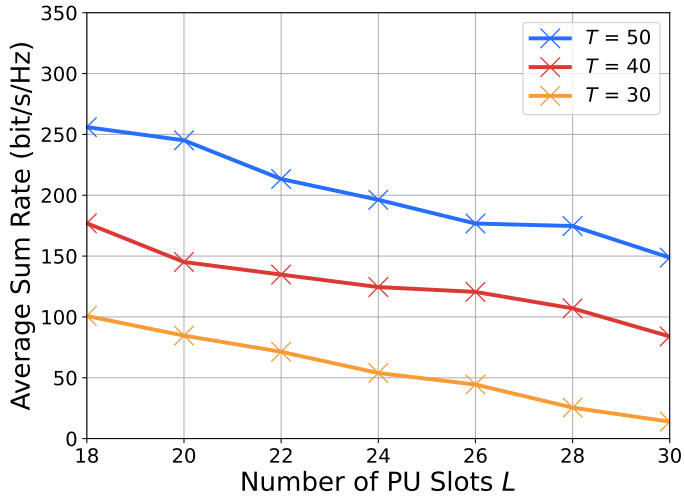


Figure 6.15: Illustrating the effect of varying the number of slots occupied by PU L and the number of time slots T on our proposed DDQN-UCB strategy.

the number of time slots T in each episode is increased, the ASR improves. This is because the CIoT agent gains more opportunities for transmission or energy harvesting, leading to an overall increase in the achieved ASR.

In Fig. 6.16, we analyze the impact of varying the initial battery level B_0 and the duration of each time slot τ on the ASR achieved by the CIoT agent using our proposed DDQN-UCB strategy. As observed, increasing B_0 leads to a higher ASR, which can be attributed to the CIoT agent facing fewer penalties, particularly in the early time slots when it has a higher battery capacity. Moreover, the figure highlights that as τ increases, the ASR also rises. This is because longer time slots provide more time for energy accumulation during the energy harvesting period $\rho_t\tau$, enhancing the CIoT agent's ability to transmit data in subsequent slots. Additionally, the transmission period $(1-\rho_t)\tau$ benefits from a longer duration, allowing for the transmission of more data. Overall, the consistent convergence of our UCB-driven DRL strategy across different scenarios demonstrates its potential to effectively enhance CIoT system performance in various settings.

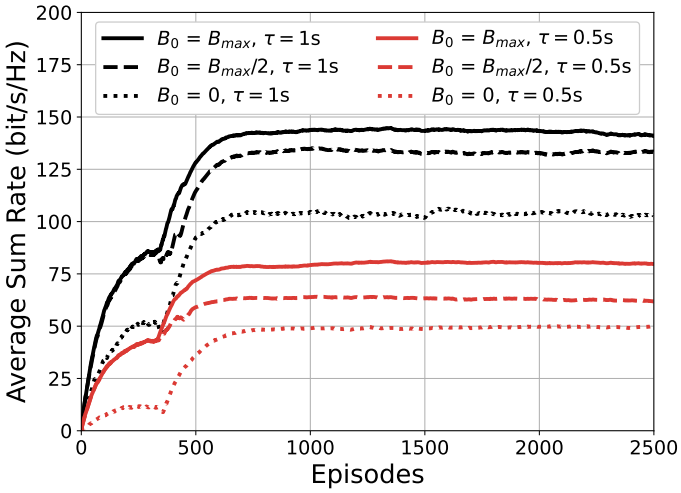


Figure 6.16: Presenting the impact of varying the initial battery level B_0 and the duration of each time slot τ on our proposed DDQN-UCB strategy.

6.6 Conclusions

In this chapter, we advanced the application of DRL for intelligent control in CIoT networks by introducing two innovative frameworks tailored for WPT-enabled and SWIPT-enabled systems. Our first contribution involved developing a system model for a WPT-enabled CIoT network with multiple competing users, where a battery-operated Tx must intelligently balance energy harvesting and data transmission. To optimize the long-term achievable sum rate under constraints, we introduced a novel DQN-based DRL approach. Expanding on this, we introduced a more flexible system model for SWIPT-enabled CIoT networks, where both transmit power and time allocation were dynamically optimized. To address this challenge, we designed a lightweight DDQN algorithm that simultaneously learns an optimal time-switching ratio and adaptively adjusts transmit power. Additionally, we integrated a UCB exploration strategy to refine decision-making under uncertainty, ensuring efficient resource utilization while mitigating interference. Comprehensive simulations validated the effectiveness of our proposed DRL solutions, demonstrating their superiority over existing benchmark approaches in terms of achievable sum rate, learning stability, and interference management. Our results highlight the transformative potential of DRL-driven techniques in intelligent spectrum sharing, adaptive power control, and energy-efficient communications in the evolving landscape of cognitive IoT.

Chapter 7

Navigating Hostile Spectrum Sharing Environments

7.1 Introduction

Cognitive IoT networks are inherently susceptible to jamming attacks due to the broadcast nature of radio wave propagation [100]. As noted in [101], advancements in software-defined radio have increased the accessibility of jamming attacks, underscoring the urgent need to protect wireless networks from both intentional and unintentional interference. In such scenarios, a jammer can disrupt communication by transmitting continuous jamming signals or short pulses over one or multiple frequency bands, thereby degrading the SNR [101]. This interference results in diminished throughput capacity and, in extreme cases, a complete disruption of transmission [102]. Consequently, jamming attacks present a significant challenge to CIoT transmissions, especially within the constraints of energy efficiency and spectrum sharing. To address these challenges, robust and intelligent countermeasures are crucial not only for enhancing CIoT network performance but also for improving security and extending operational lifetime. In this chapter, we introduce an innovative strategy for a battery-powered CIoT Transmitter (Tx) that enables autonomous decision-making to maximize long-term network throughput under spectrum-sharing constraints, mitigate jamming interference, and extend network lifespan. The proposed approach empowers the CIoT Tx to actively counter jamming attacks within the same channel.

7.2 Related Works

Although game-theoretic strategies have been investigated for anti-jamming communications [103–106], they often depend on impractical assumptions, such as prior knowledge of the jamming pattern, and may struggle against adaptive jamming tactics [107]. As a result, intelligent algorithms have gained attention in anti-jamming communications due to their ability to introduce unpredictability for jammers by dynamically adjusting to the spectrum's state [108]. By leveraging Reinforcement Learning (RL) algorithms, jamming behaviors can be inferred through a "trial-and-error" learning process within the environment, even without explicit information about the jammer. The works of [109–114] have tackled jamming attacks in the spectrum domain by employing learning-based frequency hopping techniques. The authors of [109] introduced an energy-efficient Dueling Deep Q-Network (D3QN) for implementing an anti-jamming frequency hopping strategy in CIoT networks. In [110], a convolutional Double Deep Q-Network (DDQN) was utilized to enable smart channel selection as a defense against jamming attacks. The study in [111] proposed a convolutional DDQN framework designed to mitigate interference and jamming in wideband spectrum while reducing computational complexity.

Previous studies have primarily focused on mitigating jamming attacks under the assumption that multiple channels are available and that jammers can target all channels simultaneously. Consequently, cognitive users can either select channels with a lower probability of being jammed or switch to an alternative channel when interference is detected. However, in certain scenarios, CRs are constrained to opportunistically transmit data over a certain channel. In such cases, power control strategies have been investigated as an alternative approach to counter jamming in wireless communications. The work in [115, 116] explored power control schemes that assess channel conditions and adjust transmit power to mitigate jamming effects. In [115], the authors introduced a dynamic anti-jamming model based on Q-learning to determine the optimal anti-jamming power in situations where users lack prior knowledge of the game model. While Q-learning has been applied for power control, it suffers from prolonged convergence times when dealing with a large number of states and may occasionally fail to converge [116].

Deep Reinforcement Learning (DRL), which integrates RL with Deep Learning (DL), has been

employed to address the limitations of Q-learning. The authors of [117] proposed a DDQN-based approach to develop an efficient communication policy that manages both channel access and transmission power adjustments to counter various jamming scenarios. Similarly, the work in [102] introduced a transformer encoder-based DDQN to facilitate channel and transmit power selection for a secondary transmitter operating in the presence of a jammer. In [118], a DDQN was trained using clear channel assessment data to enable a CR agent to dynamically switch channels and select optimal transmit power as part of an anti-jamming strategy. The study in [119] presented a multitask DQN framework tailored for multi-agent environments, aiming to maintain the required Quality of Service (QoS) by dynamically adjusting transmit power and frequency hopping across a wideband spectrum. Meanwhile, the authors of [116] introduced a convolutional DQN for power control in CIoT networks under jamming conditions, evaluating its effectiveness in real-world scenarios with hardware constraints.

The work in [120] investigated a power control strategy to mitigate jamming attacks in CR networks. However, the proposed cooperative mechanism may not be well-suited to the dynamic nature of CIoT networks, where users frequently join and leave on an ad-hoc basis. Additionally, it assumes a unified objective among users, which may not always hold in practice. To date, only a limited number of studies, such as [109, 116], have explicitly examined anti-jamming techniques tailored for CIoT networks, highlighting a significant research gap in this field. The authors of [109] proposed a DRL algorithm with a focus on energy efficiency, aiming to develop a system architecture that optimizes energy consumption. However, we argue that prioritizing energy-efficient DRL algorithms alone may not fully capture the broader energy constraints, especially in scenarios involving battery-powered CIoT devices. Furthermore, the approach in [116], which counters jamming by increasing CIoT device transmit power, poses significant challenges in energy-constrained environments. This issue is particularly pronounced in underlay CR settings, where secondary users share the spectrum with primary users. Studies such as [109, 116, 121] have largely disregarded spectrum-sharing scenarios in CR networks. Moreover, [116, 121] fail to consider the effects of channel fading, further emphasizing the need for a more comprehensive solution.

7.3 Contributions

To the best of our knowledge, no previous research has considered designing an intelligent algorithm that aims to maximize the CIoT network's throughput under interference limits, energy constraints, and jamming attacks. Our main contributions are, therefore:

- We introduce a novel strategy for a battery-powered CIoT transmitter, enabling autonomous decision-making to enhance long-term network throughput within spectrum-sharing limits, mitigate jamming interference, and extend network life. Our method uniquely positions the CIoT transmitter to counter jamming attacks directly in the same channel. Additionally, we evaluate the influence of small-scale fading and implement an effective Energy Harvesting (EH) model, allowing the CIoT transmitter to exclusively harvest energy from active radio frequency transmissions without dedicating infrastructure for charging.

- We formulate the throughput optimization problem for the CIoT transmitter while taking into account factors such as channel occupancy, jamming attacks, channel gain, energy arrival, battery limits, and interference constraints. This approach directs the power control and transmission decisions in the CIoT network, modelled as a model-free Markov Decision Process (MDP).

- We develop a novel DRL algorithm to learn the optimal transmission strategy that maximizes throughput without prior knowledge about the channel or jamming patterns. Our algorithm uses a DDQN designed to enable faster convergence and enhance the algorithm's energy efficiency. Additionally, we introduce an innovative Upper Confidence Bound (UCB) strategy, named UCB interference-aware (UCB-IA), meticulously designed to efficiently mitigate jamming interference and optimize the decision-making framework within the CIoT environment.

- We offer an analysis of convergence and performance of the proposed DRL algorithm that is benchmarked against alternative methodologies found in existing literature across various test scenarios. For performance evaluation, we utilize metrics such as average sum rate, average achievable reward, and jammer interference ratio. Simulations show that under the presence

of jamming attacks, the proposed learning algorithm can dynamically choose between data transmission and EH and perform power control to find the optimal solution for the network.

7.4 System Model

7.4.1 Cognitive IoT Network

Consider the spectrum-sharing CIoT network depicted in Fig. 7.1, which consists of a CIoT Transmitter-Receiver (Tx-Rx) pair. The Tx is powered by a rechargeable battery and has Wireless Power Transfer (WPT) Energy Harvesting (EH) capabilities. The CIoT network shares its spectrum with the primary network, which includes a primary Tx-Rx pair. The CIoT Tx has the ability to autonomously and dynamically adjust its transmit power P_s^t. The CIoT system operates in a time-slotted fashion, with T time slots, each lasting τ seconds. For the CIoT Tx, each time slot t consists of two phases: the decision-making phase and the operation phase. During the decision-making phase, the CIoT device decides whether to transmit data or harvest energy according to a defined policy. The decision, represented by d_t, is set to 0 when transmitting messages and to 1 when harvesting energy. Consequently,

$$d_t = \begin{cases} 0 & \text{Data transmission mode in time slot } t, \\ 1 & \text{Energy harvesting mode in time slot } t. \end{cases} \tag{7.1}$$

The operation phase involves the CIoT device executing the chosen decision d_t. It is assumed that the primary user (PU) Tx uses L slots, where $1 < L < T$, and can continuously transmit at a power level of P_p^t in each slot.

The spectrum-sharing constraint permits the CIoT Tx to operate within the same time slot as the PU Tx, as long as the interference remains below the threshold I_{th}. Thus, the CIoT device must determine its transmission power P_s^t to ensure compliance with the interference constraint, expressed as

$$P_s^t g_{sp}^t \leq I_{th}, \tag{7.2}$$

where g_{sp}^t is the channel power gain between the CIoT Tx and PU Rx. The PU status indicator is

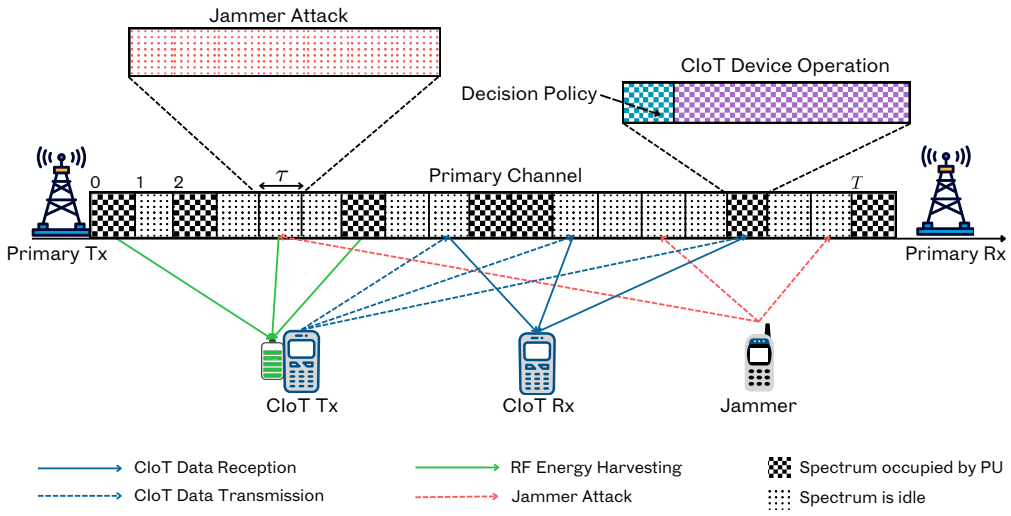


Figure 7.1: The system model of the studied CIoT network under jamming attacks and spectrum-sharing constraints.

defined as

$$\omega_p^t = \begin{cases} 1 & \text{if the PU Tx is using time slot } t, \\ 0 & \text{otherwise.} \end{cases} \tag{7.3}$$

The power gain of the channel between the CIoT Tx-Rx pair is g_{ss}^t, the channel between the PU Tx and the CIoT Rx is g_{ps}^t, and the channel between the CIoT Tx and the PU Rx is g_{sp}^t. These channels are modeled as Rayleigh fading channels that are independently and identically distributed (i.i.d.). It is assumed that the channel power gains remain constant within each time slot, but they may vary independently from one time slot to the next.

7.4.2 Jamming Model

We consider a jammer that targets the CIoT network with jamming attacks, as illustrated in Fig. 7.1. The objective of the jammer is to make the shared spectrum appear "busy", thereby preventing the CIoT Tx from accessing it and potentially draining the CIoT device's battery. It is assumed that the jammer can only target the CIoT transmissions, which may be due to the severe penalties faced by attackers if identified by the PUs, or because the jammer cannot approach the PUs. Furthermore, practical methods such as cyclostationary detection or matched filter detection

can be employed by jammers to distinguish between the PU and CIoT transmissions. Thus, at the start of each time slot, the jammer determines whether the PU is active before initiating the jamming attack. If the jammer decides to launch an attack, it will continue for the entire duration of the time slot [122]. The jammer executes the attack with power P_j^t, while the CIoT Tx has no prior knowledge of which time slots will be subject to jamming attacks.

Attackers often favor a random jamming strategy, as it allows for intermittent periods of inactivity [122]. This approach not only extends the jammer's operational lifespan but also reduces the likelihood of detection. Hence, we consider the scenario of intermittent jamming, where the jammer alternates randomly between periods of active jamming and rest. Specifically, the jammer conducts attacks for a duration of $\zeta \sim U(0, \zeta_{\max})$ slots, followed by a rest period lasting $T - \zeta$ slots, where $\zeta_{\max}$ represents the maximum number of slots the jammer can maintain attacks. Consequently, the probability of initiating an attack is given by $U(0, \zeta_{\max})/T$. By introducing randomness in the jammer's actions, the jamming behavior becomes less predictable for the CIoT Tx. In this study, we assume that the effect of a single jammer launching attacks is equivalent to multiple coordinated jammers targeting specific time slots for their attacks. This framework makes our system scalable to accommodate several coordinated jammers, each executing a jamming attack in separate time slots.

The CIoT agent determines the jammer's status indicator at the beginning of each time slot using broadband sensing capabilities [122]. The jammer's status indicator, which indicates whether a jamming attack is being launched during time slot t, is defined as follows

$$\omega_j^t = \begin{cases} 0 & \text{Jammer is launching an attack at time slot } t, \\ 1 & \text{otherwise,} \end{cases} \tag{7.4}$$

where $\omega_j^t = 0$ indicates that the CIoT agent detects the presence of a jamming attack, with a probability of $U(0, \zeta_{\max})/T$, and $\omega_j^t = 1$ signifies that the CIoT agent does not detect a jamming attack, with a probability of $1 - U(0, \zeta_{\max})/T$.

7.4.3 Energy Harvesting Model

The CIoT Tx is capable of charging its finite battery B_{max} through WPT EH [1]. At the onset, the harvested energy is set to zero. It is assumed that the amount of energy collected in each time slot, e_t, follows a uniform distribution ranging from 0 to E_{max}, where $e_t \sim U(0, E_{max})$. However, the maximum energy that can be harvested, E_{max}, is influenced by the radio frequency signals, which are influenced by the activity of both the jammer and the PU Tx during each time slot. Consequently, the value of E_{max} varies depending on the following

$$E_{max} = \begin{cases} P_p^t & \text{if } \omega_p^t = 1, \text{ PU transmitting,} \\ P_j^t & \text{if } \omega_j^t = 0, \text{ jammer is attacking,} \\ P_p^t + P_j^t & \text{if } \omega_p^t = 1 \text{ and } \omega_j^t = 0. \end{cases} \tag{7.5}$$

The energy collected in each time slot is stored in the battery and is exclusively used for operations in future time slots. However, due to hardware limitations, the CIoT Tx cannot perform EH and access the spectrum opportunistically simultaneously. It is important to note that the energy harvesting process does not result in any additional energy consumption for PUs or other devices.

The starting battery level of the CIoT Tx is denoted as B_0, with B_t representing the energy available in the battery at the t-th time slot. Following the framework in [84], we assume the battery to be ideal, meaning there are no losses of energy during storage or retrieval. For the CIoT Tx, energy consumption occurs solely due to data transmission. Furthermore, any harvested energy that is beyond the battery's capacity is discarded. The concept of normalized time slots is also employed, which allows treating "energy" and "power" synonymously [84]. At any given time slot t, the selected transmission power P_s^t by the CIoT Tx must not exceed the battery's available energy, B_t. That is,

$$0 \leq (1 - d_t) P_s^t \tau \leq B_t, \tag{7.6}$$

where τ is the duration of each time slot. The change in the battery's energy level is determined by the CIoT device's decision d_t to either harvest energy or transmit data at time slot t. In the following

[1] The implementation of circuits responsible for the process of radio frequency EH is beyond the scope of this work.

time slot, $t+1$, the available energy is updated based on d_t. Therefore, the battery update is given by:

$$B_{t+1} = \min\{B_t + d_t e_t - (1-d_t)P_s^t \tau, B_{max}\}. \tag{7.7}$$

Consequently, the total energy consumed by the CIoT device cannot exceed the total energy collected in the battery. That is,

$$\sum_{t=1}^{k} P_s^t \tau \leq B_0 + \sum_{t=0}^{k-1} e_t, \forall k. \tag{7.8}$$

where k represents the total number of time slots in which the CIoT device decides to transmit.

7.4.4 Transmission Model

The CIoT Tx is responsible for adjusting its transmit power P_s^t to maximize its total rate while under jamming attacks, ensuring it does not cause interference to the licensed network. During an idle t-th time slot, the achievable sum rate by the CIoT Tx is

$$R_0^t = \log_2\left(1 + \frac{P_s^t g_{ss}^t}{\sigma^2}\right), \tag{7.9}$$

where σ^2 represents the variance of the channel noise. If the PU Tx occupies the channel during the t-th time slot, the CIoT Tx's achievable sum rate will be reduced due to interference from the PU, as given by

$$R_1^t = \log_2\left(1 + \frac{P_s^t g_{ss}^t}{P_p^t g_{ps}^t + \sigma^2}\right). \tag{7.10}$$

Hence, the achievable sum rate for the CIoT Tx during a time slot t can be expressed as

$$R^t = \omega_j^t (1-d_t)\left[(1-\omega_p^t)R_0^t + \omega_p^t R_1^t\right]. \tag{7.11}$$

According to (7.11), if the jammer is launching an attack during the t-th time slot, i.e., $\omega_j^t = 0$, the achievable sum rate R_t of the CIoT Tx will be reduced to zero.

7.5 Optimization Problem Formulation

In this section, we aim to define the optimization problem and develop a dynamic algorithm that maximizes the total sum rate of the studied CIoT network. The algorithm will navigate challenges such as jamming attacks, channel fading, interference, and energy constraints. The CIoT device must learn to optimize both its transmission power P_s^t and decision d_t to maximize the network's throughput. Specifically, the CIoT device must strategically decide whether to transmit data, which consumes battery and may be affected by jamming, or to harvest energy, thereby forgoing immediate transmission opportunities. The optimization problem can therefore be formulated as follows

$$\max_{d_t, P_s^t} \sum_{t=1}^{T} \omega_j^t (1 - d_t) \big[(1 - \omega_p^t) R_0^t + \omega_p^t R_1^t \big], \tag{7.12a}$$

$$\text{s.t.} \sum_{t=1}^{k} P_s^t \tau \leq B_0 + \sum_{t=0}^{k-1} e_t, \ \ \forall k, \tag{7.12b}$$

$$0 \leq (1 - d_t) P_s^t \tau \leq B_t, \ \ d_t \in I \triangleq \{0, 1\}, \tag{7.12c}$$

$$d_t = 1, \ \ \forall \omega_j^t = 0, \tag{7.12d}$$

$$\omega_p^t g_{sp}^t P_s^t \leq I_{th}^t, \ \ \omega_p^t \in \Omega \triangleq \{0, 1\}. \tag{7.12e}$$

Constraint (7.12b) ensures that the transmission power P_s^t of the CIoT device remains within the limits defined by the initial battery level B_0 and the energy harvested over all time slots during the period. Constraint (7.12c) guarantees that the CIoT device's transmission power does not exceed the available energy in the current battery, B_t, during any specific time slot t. Constraint (7.12d) ensures that the CIoT device avoids transmission during time slots when the jammer is active. Furthermore, constraint (7.12e) mandates that the CIoT device's transmission adheres to the interference threshold I_{th}, preventing interference with the primary user (PU) during each time slot t. In the following discussion, we present a solution to the optimization problem defined in (7.12a) by using a model-free Markov Decision Process (MDP).

7.5.1 The Model-Free Markov Decision Process

The process of learning the optimal strategy to maximize the throughput of the CIoT network can be framed as an MDP. The MDP is represented by the tuple $(\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{P}}, \boldsymbol{\mathcal{R}}, T)$, where $\boldsymbol{\mathcal{S}}$ denotes the set of possible states in the CIoT environment, $\boldsymbol{\mathcal{A}}$ represents the set of actions available to the CIoT agent [2], $\boldsymbol{\mathcal{P}}$ represents the set of state transition probabilities, $\boldsymbol{\mathcal{R}}$ includes the rewards associated with state-action pairs, and T is the time step. In practical CR scenarios, it is challenging to accurately determine the PDF of energy and channel fading [6]. Furthermore, when under jamming attacks, calculating transition probabilities would require precise knowledge of the jammer's behavior. Since the jammer's aim is to disrupt the transmissions of the CIoT network, it intentionally does not reveal its information, making the accurate estimation of state transition probabilities unfeasible. To address this challenge, a model-free MDP approach is adopted. In this case, Deep Reinforcement Learning (DRL) is employed to infer $\boldsymbol{\mathcal{R}}$ from $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{A}}$, without requiring knowledge of $\boldsymbol{\mathcal{P}}$. As a result, the CIoT device is trained to learn a policy $\pi : \boldsymbol{\mathcal{S}} \rightarrow \boldsymbol{\mathcal{A}}$ through continuous interaction with the environment, thereby identifying the actions that yield the highest cumulative reward. This leads to a modified model-free MDP structure: $(\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{R}}, T)$. The components of this revised MDP tuple are further described below.

State Space $\boldsymbol{\mathcal{S}}$: In each time slot, the CIoT device, acting as a learning entity, evaluates the state of the unknown environment (channel) to inform its decision-making process. The state space consists of all potential states across the T time slots. For any given environment state s_t, the CIoT agent must take into account several factors: the current battery level B_t, the energy harvested in the previous time slot e_{t-1}, the presence of a primary user (PU) transmitter, the presence of jamming attacks, and the channel power gains denoted as g_{ps}^t, g_{sp}^t, and g_{ss}^t. Consequently, the state of the CIoT environment at the t-th time slot is described by these components as follows

$$s_t = \{B_t, e_{t-1}, \omega_p^t, \omega_j^t, g_{ps}^t, g_{sp}^t, g_{ss}^t\}. \tag{7.13}$$

Action Space $\boldsymbol{\mathcal{A}}$: The action space includes all possible actions the CIoT agent can take. To optimize throughput, both the decision d_t and the transmit power P_s^t should be considered as integral

[2] In this context, "ciot agent/device" refers to the transmitter in the studied network.

components of the action. Given the environment state s_t, the CIoT agent must decide whether to transmit data ($d_t = 0$) or harvest energy ($d_t = 1$), and set the transmission power P_s^t accordingly. Therefore, the action taken by the CIoT agent at time slot t is defined as

$$a_t = [d_t, P_s^t], \text{ where } d_t \in I \triangleq \{0, 1\}, \text{ and } P_s^t \in P. \tag{7.14}$$

P is the set of possible transmission powers by the CIoT agent.

Reward $\mathcal{R}$: The CIoT agent assesses the effectiveness of its chosen actions based on the acquired rewards, using this feedback to adjust its decision-making strategy. The reward is determined by the data transmission rate achieved by the CIoT Tx, provided it adheres to the constraints specified in (7.12). If the agent opts to harvest energy, the reward is set to 0. If the CIoT Tx performs an action a_t that violates the constraints in (7.12), a negative reward is assigned as a penalty. Therefore, the reward r_t for the CIoT agent at time slot t is defined as

$$r_t = \begin{cases} R_0^t & d_t = 0, \omega_p^t = 0, \omega_j^t = 1, 0 \leq P_s^t \tau \leq B_t, \\ R_1^t & d_t = 0, \omega_p^t = 1, \omega_j^t = 1, 0 \leq P_s^t \tau \leq B_t, P_s^t g_{sp}^t \leq I_{th}, \\ 0 & d_t = 1, P_s^t \tau > B_t, \\ -\phi & \text{others.} \end{cases} \tag{7.15}$$

Time Step T: The transition from the current time slot t to the next slot $t + 1$ represents a discrete time step. In this setup, we consider all possible state-action pairs over the span of T time slots, evaluating each pair systematically as the time slots progress. This iterative process ensures that the CIoT agent can adjust its actions based on the evolving environment, optimizing its strategy as it interacts with the system throughout the time steps.

7.6 DRL-Driven Throughput Optimization Under Malicious Jamming

In the context of the model-free MDP, the CIoT agent must determine the value of state-action pairs without having direct access to the state transition probabilities, $\mathcal{P}$. To address this, the CIoT agent leverages Reinforcement Learning (RL) to approximate the state-value function. By interacting

with the environment, the agent can learn an optimal policy π that dictates the selection of actions a_t based on the current environment state s_t. The goal is to maximize the cumulative reward, which in this case corresponds to the sum rate, by making informed decisions despite the challenges posed by malicious jamming, spectrum sharing requirements, energy constraints, fluctuating energy arrivals, and unpredictable channel conditions. Through this approach, the CIoT agent adapts its strategy to optimize network throughput over time.

A policy π is a function $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maps states to actions. We refer to the execution of policy π when, in state s, the corresponding action taken is $a = \pi(s)$. In the model-free MDP, the expected value of the state-action value function, often referred to as the Q-function or Bellman equation, can be expressed as

$$Q^{\pi}(s_t, a_t) = \mathbb{E}\Big[r_t + \gamma \max_{a} Q^{\pi}\big(s_{t+1}, a\big)|s_t, a_t\Big], \tag{7.16}$$

where r_t denotes the immediate reward for taking action a_t in state s_t. The term $\gamma \max_a Q^{\pi}(s_{t+1}, a)$ represents the discounted expectation of future rewards, with γ being the discount factor between 0 and 1, determining the weight given to future rewards in comparison to immediate ones. Larger values of γ give greater importance to long-term rewards. The goal of the CIoT agent is to determine the optimal action a_t that maximizes the Q-value at each time slot t.

Using the Q-learning algorithm, the CIoT agent calculates the Q-value at each step and stores it in a Q-table to find the optimal solution. The fundamental approach for updating the action-value function is outlined in [123] is

$$Q^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) + \eta\Big[r_t + \gamma \max_{a} Q^{\pi}(s_{t+1}, a) - Q^{\pi}(s_t, a_t)\Big], \tag{7.17}$$

where $\eta \in [0, 1]$ denotes the learning rate. However, Q-learning can face slow convergence when determining the optimal actions for solving the problem [121]. To address this, we explore deep Q-learning, combining concepts from both Reinforcement Learning (RL) and Deep Learning (DL), to estimate the Q-value function using a deep neural network, commonly referred to as a Double Deep Q-Network (DDQN). This method aims to improve the approximation of the Q-value function for more efficient training. A comprehensive illustration of the proposed DRL algorithm, along

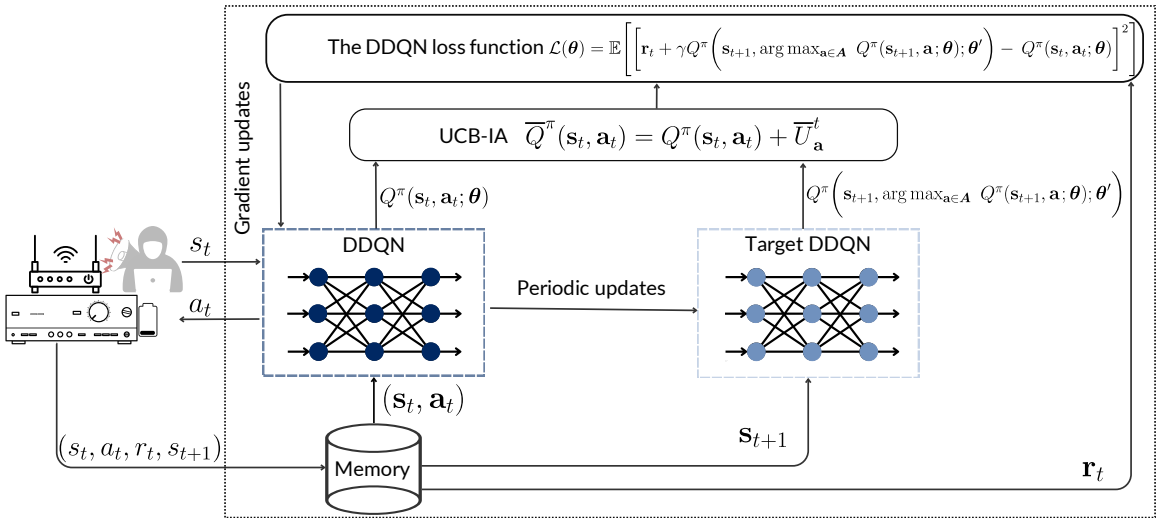


Figure 7.2: The proposed DRL algorithm featuring the UCB-IA action exploration strategy.

with our UCB-IA exploration strategy, is shown in Fig. 7.2, which will be further discussed in Subsection 7.6.2.

7.6.1 The proposed DDQN-Driven DRL Approach

In this subsection, we introduce our novel DDQN architecture, developed to identify the optimal policy for improving transmission efficiency in the CIoT network amidst malicious jamming attacks. The goal of our DDQN is to estimate the total expected reward (i.e., the Q-value) for each possible action a_t in a given state s_t. This is achieved by iteratively adjusting the DDQN parameters $\boldsymbol{\theta}$ to ensure that

$$Q^{\pi}(s, a; \boldsymbol{\theta}) \approx Q^{\pi}(s, a). \tag{7.18}$$

The DDQN parameters, $\boldsymbol{\theta} = \{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}$, represent the weights and biases of the network's layers, where $i = \{1, \ldots, 4\}$. The DDQN utilizes a fully connected neural network architecture. The input layer consists of j neurons, corresponding to the dimensionality of the state space $\boldsymbol{\mathcal{S}}$. Furthermore, the network has two hidden layers, with h_1 and h_2 neurons, respectively. The output layer is made up of z neurons.

To accelerate the convergence of the DDQN and improve the stability of the training process, we apply a weight initialization method known as Kaiming (He) initialization [95]. This approach initializes the DDQN's weights by sampling from a Gaussian distribution $\mathcal{N}(0, \frac{2}{\nu_i})$, where ν_i

represents the number of input neurons for each layer i. To introduce non-linearity into the neural network, we use the Rectified Linear Unit (ReLU) activation function. The basic ReLU function is defined as $f(x) = \max(0, x)$, which offers computational benefits over other activation functions like sigmoid or hyperbolic tangent. This is due to its simple thresholding at zero, which speeds up the training process significantly. However, the ReLU function can suffer from the "dying ReLU" problem (where neurons stop activating). To address this issue, we employ a leaky ReLU function, ensuring that all neurons remain active during the learning process, which promotes faster convergence and improves learning dynamics. The leaky ReLU activation function is defined as

$$f(x) = \begin{cases} x, & \text{if } x \geq 0, \\ \alpha x, & \text{if } x < 0, \end{cases} \tag{7.19}$$

where $\alpha \in \{0, 1\}$ represents the "slope" parameter that controls the degree of "leakiness" in the ReLU function.

To improve the training process of the CIoT agent and increase data efficiency, we implement experience replay. This approach utilizes a memory buffer $\mathcal{M}$ with a capacity of m to store previous experiences in the form of tuples (s_t, a_t, r_t, s_{t+1}). When the memory reaches its capacity, mini-batches of experiences are randomly sampled from the stored state-action pairs and used to update the DDQN's parameters. By employing this technique, we effectively reduce temporal correlations in the training data, which helps minimize the risk of instability during the training process [83].

DRL is known for its instability, and it may even diverge due to the use of a non-linear DDQNs for approximating the Q-function. Several factors contribute to this instability. Small changes in the Q-function can significantly alter the policy, which in turn affects the data distribution. Moreover, the interdependence between action-values and target values, which are derived from maximizing Q^{π} over all possible actions in the next state, exacerbates the instability. To address this issue, we employ a Target DDQN during training. The Target DDQN is used to compute the target optimal Q-function as

$$Y = \mathbf{r}_t + \gamma Q^{\pi}\left(\mathbf{s}_{t+1}, \arg\max_{\mathbf{a} \in \boldsymbol{A}} Q^{\pi}(\mathbf{s}_{t+1}, \mathbf{a}; \boldsymbol{\theta}); \boldsymbol{\theta}'\right), \tag{7.20}$$

where $\boldsymbol{\theta}'$ represents the parameters of the Target DDQN. At onset, the Target DDQN is an identical

copy of the DDQN, meaning $\boldsymbol{\theta}' = \boldsymbol{\theta}$. As training progresses, the parameters of the Target DDQN are updated at a slower rate than those of the DDQN, typically over several training steps. The rate at which the Target DDQN is updated is denoted by κ.

To update the DDQN's parameters, we utilize the Mean Squared Error (MSE) loss $\mathcal{L}(.)$ during training to quantify the deviation between the estimated Q-values and the target Q-values across a mini-batch of state-action pairs $(\mathbf{s}, \mathbf{a})$.

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}\left[\left[Y - Q^{\pi}(\mathbf{s}, \mathbf{a}; \boldsymbol{\theta})\right]^2\right]. \tag{7.21}$$

During the training phase, the objective is to minimize the loss described in (7.21) across a mini-batch of state-action pairs. This entails

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathbf{s}, \mathbf{a}). \tag{7.22}$$

The backpropagation algorithm [68] is used to compute $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathbf{s}, \mathbf{a})$, which represents the gradient of the loss function with respect to the DDQN's parameters for a batch of state-action pairs. Once this gradient is computed, Stochastic Gradient Descent (SGD) [6] can be applied to adjust the DDQN's parameters in the following manner

$$\begin{aligned} &\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathbf{s}, \mathbf{a}), \\ &\text{where } \boldsymbol{\theta} = \{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}, \\ &\text{for } i = \{1, \ldots, 4\}. \end{aligned} \tag{7.23}$$

$\eta \in \{0, 1\}$ is the learning rate that determines the update rate in each iteration of SGD.

For this study, we adopt an advanced SGD-based parameter update method called Adaptive Moment Estimation (Adam) due to its faster computation time [10, 11]. Furthermore, we implement a well-designed learning rate scheduler. At the beginning, it establishes a learning rate that promotes stable learning during the initial stages. As the training progresses, the scheduler adjusts the learning rate dynamically, considering factors like model performance and a defined patience period. This approach supports efficient convergence and improves overall performance.

7.6.2 UCB-IA: Interference-Aware Action Exploration Strategy

As widely recognized in the literature, there exists a trade-off between exploring new actions in the action space (i.e., learning their mean reward) and exploiting known actions (i.e., maximizing the empirical rewards). If the expected rewards of actions were known, the optimal policy would always select the action that provides the highest expected reward. To enable the CIoT agent to explore the environment, uncover optimal strategies, and balance the exploration-exploitation tradeoff, we apply the principles of the Upper Confidence Bound (UCB) algorithm.

The classical UCB algorithm adjusts the Q-values based on

$$\overline{Q}^{\pi}(\mathbf{s}_t, \mathbf{a}_t) = Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t) + U_a^t, \tag{7.24}$$

where U_a^t is the actual-expected reward calculated as

$$U_a^t = \hat{r}_a^t + \sqrt{\frac{c' \ln t}{C_a^t}}. \tag{7.25}$$

c' is a hyperparameter in the UCB algorithm. The actual-expected reward U_a^t is a combination of the expected reward $\hat{r}_a^t$ and an adjustment term that depends on the time period number, i.e., frame number * T + t, and the number of times action a has been selected, C_a^t. If action a_t has been selected C_a^t times by the end of time slot t (i.e., from 0 to t), then the expected reward $\hat{r}_a^t$ is calculated as $\hat{r}_a^t = (\sum_{i=1}^{C_a^t} r_{a,i}^t)/C_a^t$, where $r_{a,i}^t$ represents the reward of action a_t during its i-th selection. The action returned by the UCB algorithm is then utilized in the DDQN training, and the action count C_a^t and expected reward $\hat{r}_a^t$ are updated accordingly.

In this work, we introduce a novel variant of the UCB algorithm, referred to as Interference-Aware UCB (UCB-IA), which is presented in Algorithm 7. The proposed UCB-IA exploration-exploitation strategy not only considers the expected reward $\hat{r}_a^t$ for updating the Q-value but also incorporates the actual-expected jammer interference $\hat{\lambda}_a^t$. This modification enables the agent to enhance its performance by identifying actions affected by jammer interference in any given state s_t, thereby adjusting the Q-values to maximize the reward rate while minimizing jammer interference. As a

result, under the UCB-IA strategy, the actual-expected reward $\overline{U}_a^t$ is defined as

$$\overline{U}_a^t = \hat{r}_a^t \hat{\lambda}_a^t + \sqrt{\frac{c' \ln t}{C_a^t}}. \tag{7.26}$$

We express the actual-expected jammer interference $\hat{\lambda}_a^t$ as

$$\hat{\lambda}_a^t = \frac{\sum_{i=1}^{C_a^t} \lambda_{a,i}^t}{C_a^t}, \tag{7.27}$$

where $\sum_{i=1}^{C_a^t} \lambda_{a,i}^t$ represents the total number of times the CIoT agent has encountered a jamming attack due to action a. Therefore, $\hat{\lambda}_a^t$ takes a value between 0 and 1. Consequently, for the proposed UCB-IA algorithm, the adjusted Q-value is given by

$$\overline{Q}^{\pi}(\mathbf{s}_t, \mathbf{a}_t) = Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t) + \overline{U}_a^t. \tag{7.28}$$

This adjustment, illustrated in Fig. 7.2, is applied to the Q-value output of the DDQN architecture, enhancing the action selection policy in alignment with the proposed UCB-IA strategy.

Algorithm 7 The proposed UCB-IA-driven DRL algorithm to solve (7.12)

Input: Cognitive IoT environment simulator and its parameters.
Output: Optimal action a_t in each time slot t.
Initialize experience replay memory $\mathcal{M}$ with size m.
Initialize battery level with B_0
Initialize $\forall \theta \in \boldsymbol{\theta}, \quad \theta \sim \mathcal{N}(0, \frac{2}{v_i})$ and initialize $\boldsymbol{\theta}'$ with $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}$.
Initialize η and set the scheduler's reduction factor and patience period.
Initialize γ, κ, and c'.
for episode = 1 to episodes **do**
    for $t = 1$ to T **do**
        Observe the state s_t
        if $\mathcal{M}$ is not full **then**
            Sample a random action a_t
            Get the reward r_t using (7.15) and observe the next state s_{t+1}
            Store $\mathcal{M} \leftarrow (s_t, a_t, r_t, s_{t+1})$
        else
            Calculate $\overline{U}_a^t \leftarrow \hat{r}_a^t \hat{\lambda}_a^t + \sqrt{\frac{c' \ln t}{C_a^t}}$
            Adjust Q-value $\overline{Q}^{\pi}(s,a) \leftarrow Q^{\pi}(s,a) + \overline{U}_a^t$
            Get action a_t according to the policy of adjusted Q-value
            Update action count, $C_a^t \leftarrow C_a^t + 1$
            Get the reward r_t using (7.15) and observe the next state s_{t+1}
        end if
        Update $\hat{\lambda}_a^t \leftarrow (\sum_{i=1}^{C_a^t} \lambda_{a,i}^t)/C_a^t$
        Update $\hat{r}_a^t \leftarrow (\sum_{i=1}^{C_a^t} r_{a,i}^t)/C_a^t$
        Sample a mini-batch X from $\mathcal{M}$
        Predict Target Q-values using (7.20)
        Predict Q-values using $Q^{\pi}(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\theta})$
        Calculate the loss in (7.21)
        Update $\boldsymbol{\theta}$ of DDQN online using (7.23)
        if episode * t mod κ = 0 **then**
            Update $\boldsymbol{\theta}'$ of Target DDQN online as $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}$
        end if
    end for
    Update η using scheduler
    Update ϵ
    Update the state $s_{t+1} = s_t$
end for

7.7 Simulation Results

In this section, we assess the effectiveness of our proposed DRL strategy featuring the proposed DDQN and the UCB-IA exploration in enhancing the transmission efficiency of the EH-enabled CIoT network detailed in Section 7.4.

7.7.1 Setup

The simulation parameters used are presented in Table 7.1. For the evaluation, we use several performance metrics, including Average Sum Rate (ASR), average achievable reward, and jammer interference rate across training episodes. The total achievable sum rate of the CIoT Tx agent over T time slots (one episode) is calculated as $\sum_{t=1}^{T} R^t$, where R^t is defined in (7.11). The ASR represents the weighted moving average of the total achievable sum rate. In addition, the total reward is the sum of all rewards accumulated by the CIoT Tx agent during one episode, expressed as $\sum_{t=1}^{T} r_t$, with r_t provided in (7.15). The average reward is the weighted moving average of the total reward. Finally, the jammer interference rate is determined by the ratio of the number of time slots in which the CIoT agent transmits data while the jammer is active, to the total number of time slots under jamming attacks. This is calculated as $\sum_{t=1}^{T} \lambda^t / \sum_{t=1}^{T} \omega_j^t$, where ω_j^t is defined in (7.4) and $\lambda^t = 1$ indicates that the CIoT agent was subjected to jamming in time slot t (with $\lambda^t = 0$ otherwise).

The calculation of the weighted moving average is as follows

$$\text{average}_{\text{new}} = (1 - \delta) \times \text{average}_{\text{old}} + \delta \times \text{value}. \tag{7.29}$$

Here, δ represents the weight given to the most recent data point, with $1-\delta$ reflecting the importance of the accumulated historical average. In this study, we have set $\delta = 0.01$. The use of a weighted moving average helps to smooth out short-term fluctuations, making the underlying trends in both sum rate and rewards clearer. This approach effectively strikes a balance between incorporating new data and retaining the relevance of past information, thus improving the analysis of training episodes.

Table 7.1: Simulation parameters for the proposed EH-enabled CIoT network under jamming attacks employing our proposed DRL approach with UCB-IA exploration strategy.

Parameters	**Value**
Number of time slots T	30
Duration of each time slot τ	1 s
Number of PU transmission slots L	18
Transmission power of PU P_p^t	0.2 W
Interference threshold I_{th}	0.01 W
Initial battery level B_0	0.0 W
Battery capacity B_{max}	0.5 W
Transmission power range of CIoT Tx P_s^t	$0.01 \sim 0.1$ W
Transmission power of jammer P_j^t	0.1 W
Maximum time slots under jamming $\zeta_{\max}$	12
Noise power σ^2	1e-3 W
Experience replay memory size m	10,000
Training episodes	2500
Mini-batch size	200
Learning rate η	$4 * 10^{-4}$
Learning rate reduction factor	50%
Learning rate patience period	500 episodes
Penalization ϕ	7
Discount factor γ	0.99
Leakiness parameter α	0.02
Update rate of Target DDQN κ	100
UCB adjustment term c'	1
Channel power gain of CIoT Tx-PU Rx g_{sp}^t	0.2 W
Channel power gain of CIoT Tx-Rx g_{ss}^t	0.1 W
Channel power gain of PU Tx-CIoT Rx g_{ps}^t	0.2 W
Number of neurons	7, 128, 64, 22

To evaluate the performance of our proposed DRL approach, we compare it with the following strategies:

- The ϵ-greedy strategy, which is commonly used in the literature to balance exploration and exploitation. With the ϵ-greedy strategy, the CIoT agent selects an action to maximize the estimated Q-value (exploitation) with a probability of $1 - \epsilon$, and randomly chooses an action (exploration) with a probability of ϵ.

- The Fixed strategy, where the CIoT agent determines its action a_t at each time step using a rule-based approach derived from the constraints in (7.12), without employing any learning mechanisms [14].

- The Random strategy, in which the CIoT agent selects an action a_t at each time step randomly

from the action space, without any intelligent decision-making.

7.7.2 Results and Analysis

To ensure a fair comparison between our UCB-IA strategy and the ϵ-greedy strategy, Fig. 7.3 demonstrates how different values of ϵ influence the performance of the DRL algorithm, helping us select the optimal ϵ value to maximize the ASR. As ϵ varies, the ASR of the CIoT agent undergoes significant changes. The highest ASR is achieved when $\epsilon = 0.1$, indicating that lower ϵ values offer an ideal balance between exploration and exploitation. This enables the CIoT agent to make informed decisions based on its accumulated knowledge while still occasionally exploring new actions. On the other hand, as ϵ increases, the CIoT agent's ASR declines. The worst performance occurs at $\epsilon = 0.9$, where the agent predominantly explores, behaving almost like the random strategy. This excessive exploration leads to frequent penalties, as the CIoT agent either selects random actions or disregards its learned knowledge, resulting in suboptimal decisions and reduced performance. Therefore, we select $\epsilon = 0.1$ for the ϵ-greedy strategy to ensure a fair comparison with our proposed approach.

In Fig. 7.4, we show the ASR of the CIoT Tx over training episodes for various strategies. At the beginning of training, both the proposed UCB-IA strategy and the ϵ-greedy strategy exhibit an ASR similar to that of the random strategy, as the CIoT agent is still in the process of building its

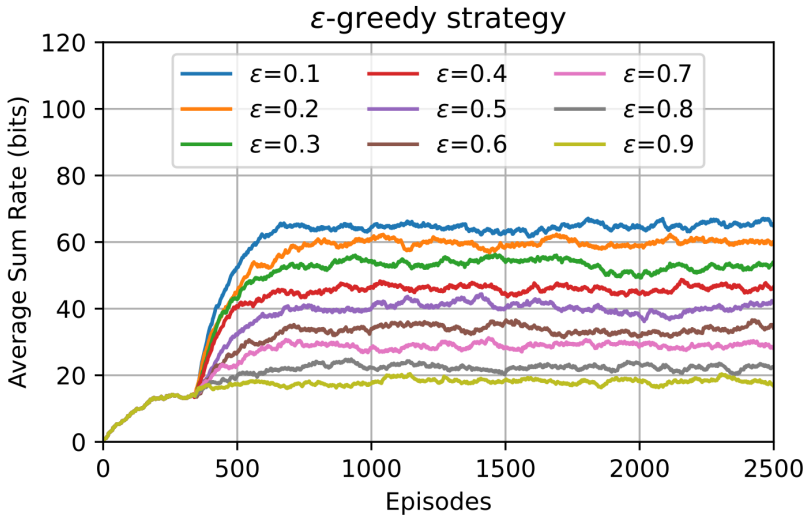


Figure 7.3: The CIoT Tx's ASR performance with ϵ-greedy strategy across training episodes, comparison of different greediness value ϵ.

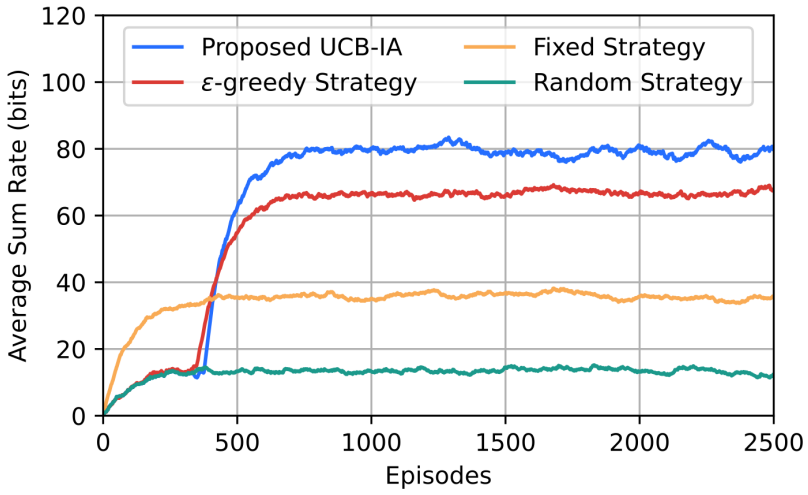


Figure 7.4: The CIoT Tx's ASR performance across training episodes, comparison of different strategies.

experience in the replay memory. However, at convergence, it becomes clear that the proposed UCB-IA strategy achieves the highest ASR when compared to all other strategies. The high performance of the CIoT agent using our DRL approach is attributed to the UCB-IA algorithm, which adjusts the Q-value to help the agent balance the trade-off between exploitation and exploration, factoring in both expected reward and jammer interference. As a result, the CIoT network can efficiently share the spectrum with the primary network while maximizing throughput under malicious jamming conditions. The ϵ-greedy strategy outperforms both the random and fixed strategies, but it still significantly lags behind the UCB-IA strategy, indicating that it is not the optimal approach for balancing the exploration-exploitation trade-off in such a dynamic CIoT environment. The random strategy results in the lowest ASR due to its random action selection, which reduces the chances of successful data transmission. The fixed strategy, which follows predefined rules, ranks second-lowest in ASR, highlighting the limitations of its suboptimal action choices, as the agent cannot explore all potential actions that could improve the ASR.

Fig. 7.5 presents the average rewards achieved by the CIoT agent over training episodes using different strategies. The figure illustrates the convergence of all learning-based strategies, validating the effectiveness of the training process. Both the proposed UCB-IA and ϵ-greedy strategies start with zero rewards at the beginning of training, as the CIoT agent is accumulating experiences in the replay buffer. Following this, both strategies experience a temporary drop in rewards, which is

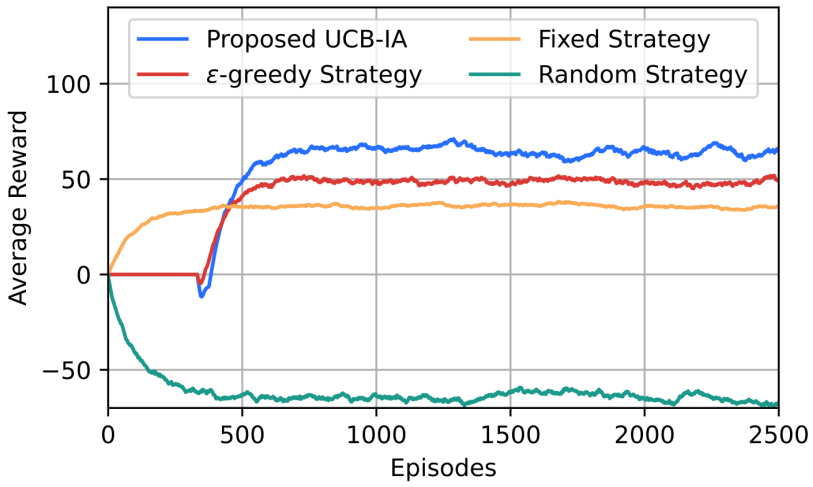


Figure 7.5: The CIoT Tx's average achievable reward across training episodes under different strategies.

followed by an increase. This drop is due to the DRL-based strategies prioritizing long-term rewards over short-term rewards as a result of exploration. Similar to the trends shown in Fig. 7.4, our proposed DRL algorithm with the UCB-IA strategy achieves the highest average reward, followed by the ϵ-greedy strategy and the fixed strategy, while the random strategy registers the lowest reward. The fixed strategy maintains positive rewards throughout all training episodes, as it follows predefined rules that avoid actions resulting in penalties. On the other hand, the random strategy consistently yields negative rewards across all episodes due to its random action selection. By comparing Fig. 7.4 and Fig. 7.5, it is evident that for both UCB-IA and ϵ-greedy, the ASR values are higher than the corresponding reward values. This is because the average reward metric considers penalties incurred during the exploration of new actions. Nonetheless, despite the penalties during exploration, the reward of our proposed UCB-IA strategy remains higher than that of the ϵ-greedy strategy.

In Fig. 7.6, we present the jammer interference rate across all training episodes for the four strategies previously discussed. The results show that the random strategy experiences the highest interference rate, approximately 50%. This indicates that, on average, the CIoT agent transmits data during about half of the time slots when jamming signals are present, leading to penalties and data loss. In contrast, the fixed strategy ensures a zero interference rate for the CIoT agent throughout all training episodes. However, as shown in Fig. 7.4, its lower performance in achieving

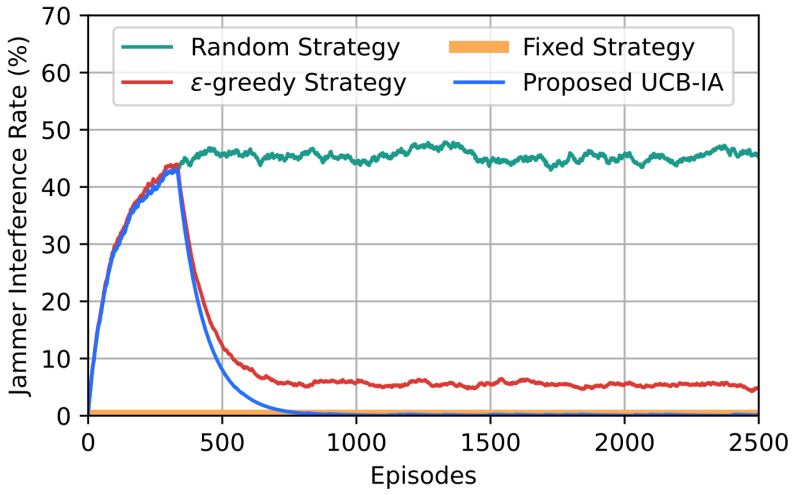


Figure 7.6: The jammer interference rate with the CIoT agent across training episodes under different strategies.

the ASR suggests that it is not the optimal approach. The zero interference rate achieved by the fixed strategy is a result of its rule-based design, which follows the constraints specified in equation (7.12), particularly constraint (7.12d). This constraint ensures that the agent switches to energy harvesting if the jammer is active during time slot t. Initially, the jammer interference rate for both the ϵ-greedy and UCB-IA strategies increases as the CIoT agent accumulates experiences. However, as training progresses, the CIoT agent using the ϵ-greedy strategy learns the actions that maximize the ASR, although those actions lead to a 5% interference rate with the jammer at convergence. In contrast, when the CIoT agent employs the proposed UCB-IA strategy, it learns actions that not only optimize long-term throughput but also result in a 0% interference rate with the jammer. This suggests that the UCB-IA strategy enables the CIoT device to effectively manage jammer interference on the same channel by harvesting energy from these interference signals, thereby improving battery levels and transmission success in subsequent time slots.

In Fig. 7.7, we show the ASR for the four strategies across different values of the maximum battery capacity $B_{\max}$ of the CIoT device. As observed in the figure, increasing $B_{\max}$ enables the CIoT device to harvest more energy, leading to an increase in the ASR. Additionally, a larger battery capacity reduces the chances of battery overflow (when harvested energy exceeds $B_{\max}$), thereby minimizing the penalties encountered by the CIoT agent. However, once a certain threshold is reached, further increases in battery size result in diminishing returns, indicating a point of

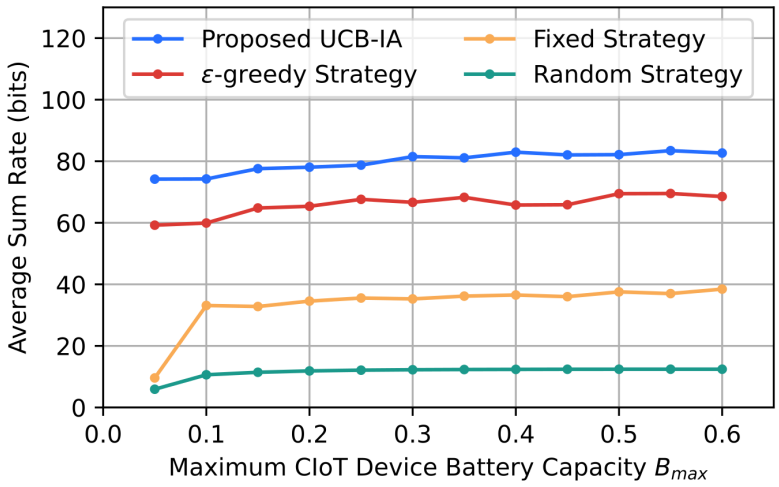


Figure 7.7: The effect of the maximum battery capacity B_{max} of the CIoT agent on the ASR across different strategies.

resource saturation. At this point, additional capacity does not translate into further improvements in performance. Despite this, our proposed DRL algorithm using the UCB-IA strategy consistently outperforms all other strategies across various values of $B_{\max}$. This highlights the adaptability of our approach in optimizing the ASR not only in scenarios with abundant battery capacity but also in situations with constrained capacity. It is worth noting that, even at higher battery capacities, the performance of the CIoT agent using the ϵ-greedy strategy remains inferior to that of our proposed method.

In Fig. 7.8, we show the ASR of the CIoT agent using our proposed DRL algorithm combined with the UCB-IA strategy across all training episodes, considering various initial battery levels B_0. The full battery configuration achieves the highest ASR among all initial battery levels, while the empty battery configuration yields the lowest ASR. This is due to the fact that starting with an empty battery restricts the CIoT agent's available actions, increasing the likelihood of selecting actions that result in penalties. Therefore, the agent focuses on energy harvesting in the initial time slots to ensure sufficient power for data transmission in subsequent slots. In contrast, starting with a full battery enables the CIoT agent to prioritize data transmission, thereby achieving higher rewards.

In Fig. 7.9, we display the ASR of the CIoT agent under different spectrum-sharing scenarios by varying the number of slots occupied by the PU Tx L. As L increases, a noticeable decrease in

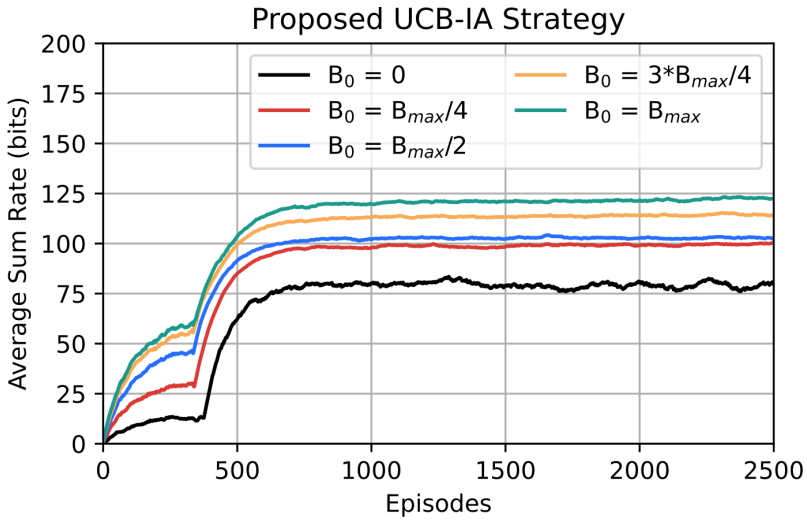


Figure 7.8: The effect of starting battery level B_0 on the ASR of the CIoT Tx using our proposed UCB-IA approach.

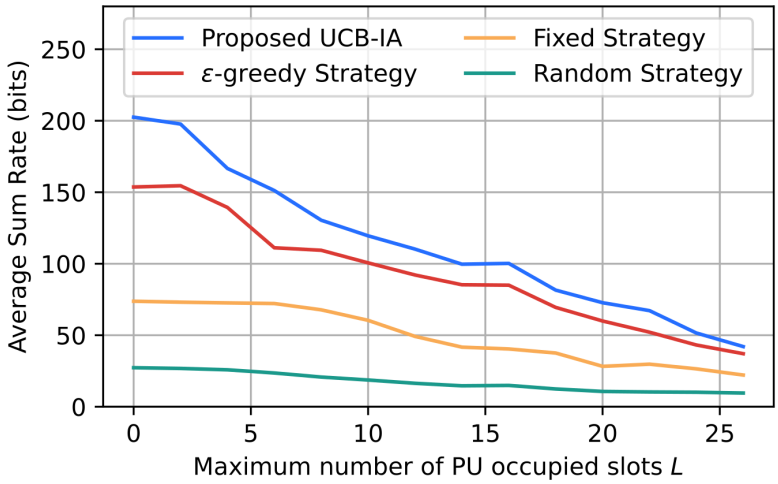


Figure 7.9: The effect of the number of PU transmission slots L on the CIoT device's ASR across different strategies.

the ASR of all strategies occurs. This reduction is due to the growing restriction on action selection by the CIoT agent as more slots are occupied. During an occupied slot, the agent faces a limitation imposed by the interference threshold I_{th}, which requires selecting lower transmit power P_s^t to avoid penalties, leading to a lower ASR. Additionally, it is evident that at high values of L, all strategies perform similarly, as the PU Tx occupies nearly all the slots. Conversely, at lower values of L, the CIoT agent has more flexibility in selecting its transmit power, resulting in a higher ASR. Despite the

variations in L, our proposed DRL algorithm with the UCB-IA strategy consistently outperforms all other strategies. This demonstrates that our approach enables the CIoT network to maximize throughput while effectively coexisting with the PU Tx, even as channel occupancy fluctuates.

In Fig. 7.10, we illustrate the ASR of the CIoT agent across various numbers of time slots T under different strategies. It is clear that as the number of time slots increases, the ASR for all strategies improves. This is because, with a fixed number of slots occupied by the PU Tx and targeted by the jammer, an increase in the number of time slots T provides the CIoT agent more opportunities for transmission without penalties. As shown in the figure, the proposed UCB-IA strategy and the ϵ-greedy strategy exhibit a more substantial increase in ASR compared to the fixed and random strategies. This difference can be attributed to the learning capabilities of these strategies, allowing them not only to select the optimal action at each time slot but also to adjust the transmit power, which directly affects the ASR. At 20 time slots, the ASR values for all four strategies converge and are nearly identical. This occurs because, at $T = 20$, there are 18 slots occupied by the PU Tx and 10 slots experiencing jamming, limiting the available actions for the CIoT agent and resulting in similar choices across all strategies. However, as T increases, the ASR of the UCB-IA and ϵ-greedy strategies surpasses that of the fixed and random strategies by a significant margin.

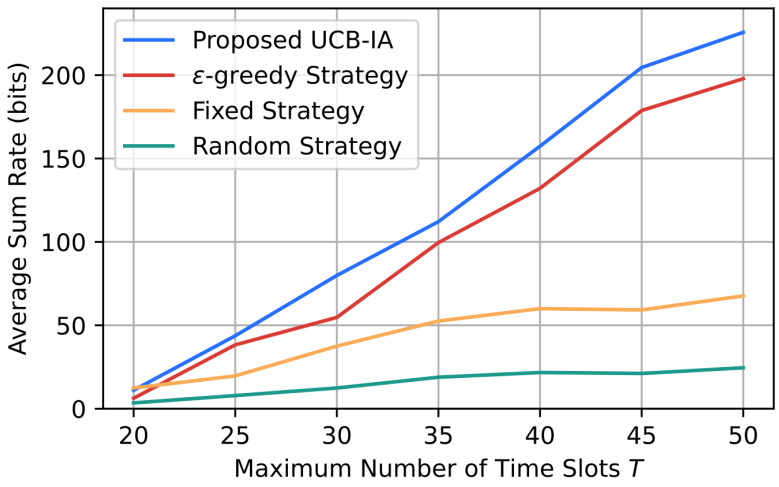


Figure 7.10: The effect of the number of time slots T on the CIoT Tx's ASR across different strategies.

7.8 Conclusions

In this chapter, we explored the potential of deep reinforcement learning to enhance the security and performance of spectrum-sharing cognitive IoT networks, particularly in adversarial radio environments impacted by jamming attacks. We proposed a novel DRL algorithm specifically designed to balance the exploitation-exploration trade-off, optimizing action selection for maximizing sum rates in hostile jamming conditions. This algorithm allows the CIoT agent to not only mitigate jamming attacks but also leverage the interference from jamming to achieve its objectives and extend its operational lifespan. Our findings demonstrate that the proposed DRL algorithm, using the UCB-IA strategy, successfully achieves its goals, significantly outperforming existing benchmarks. This underscores the importance of customizing DRL techniques to the unique dynamics of the system. We also validated the algorithm's convergence across different network conditions, confirming its potential to enhance CIoT network performance even in challenging environments.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

Next-generation wireless networks are expected to offer ubiquitous access to a wide range of communication services, supporting applications from smart cities and drone missions to augmented and virtual reality. This rapid expansion, which shows no sign of slowing, demands a reevaluation of spectrum utilization efficiency. Dynamic spectrum access, facilitated by Cognitive Radio (CR) technology, is poised to play a crucial role in these networks. However, true cognition requires the ability to learn from experience, which is why Artificial Intelligence (AI) holds immense potential in enhancing cognitive networks, elevating them to new levels of intelligence. Through this thesis, we have demonstrated that the synergistic combination of AI and CR technology enables autonomous decision-making, while providing wireless devices with *context-awareness*, *self-management*, *self-optimization*, and *self-sustaining* capabilities. These advancements empower intelligent radio devices to efficiently handle critical tasks such as dynamic spectrum access, power management, resource allocation, and security. Furthermore, we have shown that by relying solely on raw radio environment data and continuous interaction with the environment, we can develop "*brain-powered*" autonomous systems capable of intelligently invoking their perception, reasoning, and judgment capabilities.

In this thesis, we have addressed key challenges in AI-enabled spectrum-aware networks, proposing innovative AI solutions while also exploring supporting technologies such as energy harvesting,

which can contribute to the development of self-sustainable green networks. Specifically, we have focused on developing unsupervised learning approaches to drive frequency-domain context-awareness, enabling more accurate spectrum gap identification. This is crucial, as most existing approaches rely on supervised learning, which are critical in CR contexts. Additionally, we have pioneered some of the first unsupervised deep representation learning frameworks for robust spectrum data representation, enhancing perception capabilities in both large- and small-scale cooperative networks. To promote edge intelligence and improve privacy in mobile cognitive networks, we introduced the first fully unsupervised and distributed learning framework. This framework allows users to retain control over their spectrum data, eliminating the need to transmit it to a central entity for aggregation. Furthermore, we have made significant advancements in developing intelligent control algorithms for power management and resource allocation in resource-constrained Cognitive IoT (CIoT) networks, equipping nodes with the ability to autonomously adapt to dynamic, uncertain, and potentially hostile spectrum-sharing environments without requiring comprehensive prior knowledge. The research presented in this thesis has resulted in a series of publications [8–18], underscoring its substantial contributions to AI-driven spectrum-aware networks. A brief summary of these achievements is provided below:

(1) To enable AI-driven context-awareness for enhanced spectrum sensing in large-scale cooperative CR networks, we have proposed several novel unsupervised Machine Learning (ML) methods that require no labeled training data, are data-efficient, and function without relying on communication with the licensed network. We have demonstrated that our proposed approaches achieve comparable sensing performance to traditional supervised learning models that rely on labeled datasets.

- We have demonstrated how supervised models can be trained using unsupervised data, achieving superior performance without the need for labeled data.
- By applying dimensionality reduction, we enhance the computational efficiency and generalizability of unsupervised models, ultimately improving the detection performance of idle spectrum.
- We have illustrated how unsupervised learning can be leveraged to intelligently determine

the full range of licensed channel states, not just the idle/busy states. This approach, applied in hybrid interweave-underlay CR access mode, capitalizes on the performance gains that can result from accurately identifying these states.

(2) To enhance the reasoning and analysis capabilities of cognitive networks, we have made significant strides by proposing some of the first fully unsupervised deep representation learning frameworks. These frameworks efficiently learn disentangled representations of spectrum data without relying on large datasets or complex architectures. Our proposed solutions cater to both small-scale and large-scale cooperative CR networks. Extensive simulations show that, across various environment settings, the proposed approaches achieve performance comparable to supervised Deep Learning (DL)-based methods, while outperforming non-DL approaches.

- We have revealed that, in small-scale networks with a limited number of cooperating users for idle spectrum identification, deep representations can significantly boost performance in such constrained settings.
- We have effectively addressed challenges associated with unsupervised learning, such as sensitivity to cluster centroid initializations and the expected cluster count. This brings us closer to fully automatic approaches that learn directly from the data, without explicit knowledge, improving their practicality.
- We have validated that, through specialized architectures, we can not only learn disentangled representations that improve sensing performance, but also develop generative models capable of generating new, unseen examples by learning the data distribution in latent space.

(3) To improve privacy in large-scale mobile cognitive networks, we have developed the first unsupervised deep federated learning approach, enabling robust, distributed, and secure spectrum sensing. In this approach, we leverage user mobility to collect spectrum data, which is then used to collaboratively and locally train a shared deep representation learning model that performs non-linear compression.

- Due to the mobility of multiple users, each experiences different channels and environment conditions. As a result, each device collects data with significantly varied distributions. This "data heterogeneity" improves the distributed training process, leading to better model generalizability and performance.
- In our proposed approach, instead of transmitting the actual measured spectrum data, users send only the updated model parameters to the fusion center for aggregation. This significantly reduces communication overhead by avoiding the transmission of large amounts of data and also enhances user privacy, placing control back in the hands of the users.
- With this approach, a newly joining user can download the model parameters from the fusion center, enabling it to transform its collected spectrum data into a more efficient representation, thereby enhancing its judgment capabilities.

(4) To develop intelligent and adaptive control algorithms for the joint management of various network resources in resource-constrained Cognitive IoT (CIoT) networks, we have proposed two novel lightweight Deep Reinforcement Learning (DRL) approaches. These approaches are designed to autonomously learn operational strategies to optimize network resources in spectrum-sharing networks, with the goal of maximizing long-term achievable throughput. Against benchmarks, we have consistently demonstrated both the convergence of our proposed approaches and their superior performance compared to other methods.

- We have indicated that through realistic energy harvesting models, which do not rely on the presence of a stable source for recharging, devices can capitalize on spectrum energy to recharge their limited batteries, bringing us closer to self-sustaining, green networks.
- We have established that DRL algorithms enable autonomous learning through constant interactions with the dynamic environment, without requiring comprehensive prior knowledge, advancing the development of autonomous spectrum-agile networks.
- We have demonstrated that the choice of action exploration strategy can significantly improve the learning performance of the DRL algorithm, which in turn leads to higher performance gains for the network.

(5) To intelligently navigate hostile spectrum-sharing environments, we have developed a DRL-driven approach that allows a CIoT agent to confront jamming attacks directly within the same channel, without relying on frequency hopping strategies. The algorithm is designed for fast convergence, enhancing energy efficiency and ensuring rapid adaptability to adversarial conditions.

- We have demonstrated how both the proposed DRL algorithm and our novel interference-aware action exploration strategy enable the CIoT device to intelligently learn a transmission strategy that not only effectively mitigates jamming attacks but also maximizes network performance.
- We have shown how jamming attacks in these networks can be turned into a benefit for the user. When such attacks occur, the user can harvest energy from them, transforming what would otherwise be a disruptive event into a valuable opportunity.

The future of wireless networks will undoubtedly depend on AI, and the research presented in this thesis contributes to understanding the essential tools for developing a new generation of intelligent, spectrum-aware, and spectrum-agile wireless networks.

8.2 Future Work

In this section, we outline promising research directions to build upon the findings presented in this thesis.

Overcoming Labeled Data Scarcity in Wireless Communications. The scarcity of labeled data in wireless communications stems primarily from data privacy concerns and the high cost of data labeling. Additionally, spectrum availability fluctuates dynamically over time and space due to variations in the number of users, interference, and environment factors. This variability makes it challenging to obtain accurately labeled data that fully represents the range of possible scenarios. To address these limitations, synthetic data generation techniques have been employed. These techniques create model-driven artificial data that mimics real-world data while being more cost-effective and easier to generate. However, the absence of real-world labeled data can hinder the

effectiveness and generalizability of such approaches. Since synthetic data may not fully capture the complexity of real-world radio environments, models trained on it may struggle in practical applications. One way to mitigate this limitation is by constructing realistic wireless environments using laboratory equipment and curating datasets accordingly. Alternatively, environment simulators can be leveraged to virtually replicate real-world conditions. However, each approach has its trade-offs: while lab-based simulations provide valuable data collection opportunities, they may not fully reflect real-world conditions, whereas virtual simulators offer a controlled setting for data generation but may lack complete real-world fidelity. To further address data scarcity, few-shot learning can be utilized to train models with minimal labeled examples per class, while zero-shot learning enables models to recognize and classify unseen examples without prior exposure. These approaches help overcome the challenges posed by limited labeled data while improving model adaptability to real-world scenarios.

Handling Missing or Erroneous Data. Data quality is crucial for the success of AI algorithms in spectrum-aware networks, as these algorithms rely on accurate and reliable data from sensors and devices. Errors or inconsistencies in the data can lead to flawed models, reducing the network's effectiveness. Therefore, it is essential to develop AI algorithms that can handle data inconsistencies and errors while maintaining robust performance. Moreover, AI algorithms should be capable of adapting to variations in data quality over time. This can be achieved through techniques such as adaptive learning, which enables models to update their parameters in response to shifts in data distribution. In scenarios where unlicensed users join and leave the network on an ad-hoc basis, models expecting a fixed number of data features can encounter issues. Missing data may arise when users leave, or an excess of features may be introduced when new users join. While a simple approach like truncating extra features can be used to handle the latter, it reduces the model's degrees of freedom. On the other hand, missing data can create significant problems for model accuracy. Future research should focus on developing AI algorithms capable of effectively handling missing data during both training and prediction for cognition tasks.

Considering Imperfect Reporting Channels in Cooperative Networks. Cooperation among unlicensed users in spectrum sensing enhances the likelihood of detecting licensed users by aggregating spectrum data from spatially diverse users, each subjected to different channel conditions, as well

as independent shadowing and fading effects. Additionally, cooperation helps mitigate the hidden node problem, where an unlicensed user may struggle to detect signals from licensed users. This advantage, known as cooperative gain, reduces the sensitivity requirements for unlicensed users, enabling the deployment of lower-cost radio devices without the need for highly precise measurements. In cooperative networks, unlicensed users transmit their sensed spectrum data to a fusion center over independent reporting channels. While many studies assume that fading affects only the channels between PUs and SUs during local sensing, they often consider the reporting channels between SUs and the fusion center to be ideal, ensuring error-free transmission of local decisions. However, in real-world scenarios, these reporting channels also experience fading, introducing errors that can degrade the accuracy of the fusion center's global decision. Therefore, it is essential to analyze the impact of imperfect reporting channels on the performance of learning-driven cooperative spectrum sensing methods.

End-to-End Anomaly Detection Through Deep Autoencoder Architectures. The research presented in this thesis demonstrates that deep representation learning using various autoencoder architectures can significantly enhance the detection of spectrum gaps. These architectures typically consist of two major components: the encoder and the decoder. The encoder maps spectrum data onto a latent representation space, where the task of spectrum hole identification takes place. Beyond improving performance, these architectures can also serve as a defense mechanism against sensing data falsification attacks in CR networks. In such attacks, malicious users attempt to deceive the fusion center by sending misleading spectrum sensing data. One way to mitigate this threat is through end-to-end anomaly detection using autoencoder architectures. Specifically, the encoder can be deployed locally on legitimate unlicensed users' systems, while the decoder resides at the fusion center. In this setup, when data originates from a legitimate unlicensed user, the decoder can reconstruct it with minimal error. However, if an illegitimate user transmits raw, unencoded sensing data, the decoder at the fusion center will fail to properly reconstruct it, allowing the system to detect and flag the anomaly.

Designing Interpretable and Explainable AI Algorithms for Spectrum-Aware Networks. Uncertainty in AI stems from the probabilistic nature of many learning models, which provide a range of possible outcomes rather than definitive answers. This uncertainty can arise due to missing

or noisy data, model complexity, or the inherent randomness of learning algorithms. Interpretability, on the other hand, refers to the ability to understand how a model arrives at its predictions. This is particularly crucial in autonomous CR systems, where the learning model's decisions can have significant consequences. Ensuring interpretability is essential for building trust, maintaining fairness and accountability, and identifying potential biases. However, many AI models function as *black boxes*, making their decision-making processes difficult to comprehend. To address this challenge, there is growing interest in developing methods to enhance the interpretability and transparency of AI models. Graphical models, for instance, offer a structured way to represent relationships between variables, making dependencies and causal links easier to understand. Other inherently interpretable learning models include Decision Trees (DTs) and Convolutional Neural Networks (CNNs). DTs provide clear, visual decision pathways, offering transparency in model reasoning, while CNNs can leverage techniques to highlight key regions influencing predictions. However, the applicability of these models may be limited in certain problem domains. Despite ongoing advancements, uncertainty in AI remains an open challenge, and further research is needed to develop robust, interpretable solutions for complex decision-making tasks in spectrum-aware networks.

Combating Adversarial Attacks on Learning Models for Cognition Tasks. Integrating AI with CR brings significant performance improvements but also introduces security vulnerabilities. While security measures exist to protect communication layers, privacy-preserving techniques are essential to safeguard the learning algorithms themselves. Any AI model can be stolen by extracting its trained parameters or decision boundaries. A model is represented by the equation $y = f(x, w)$, where x is the input, y is the output, and w represents the model's parameters. By feeding multiple inputs into the model and recording its responses, an attacker can collect enough data to solve for w, effectively replicating the model. This vulnerability makes learning models susceptible to adversarial attacks, where carefully crafted inputs deceive the model, leading to incorrect predictions and compromising both security and performance. Beyond security threats, relying entirely on learning algorithms poses an additional risk—creating a single point of failure. If an attacker successfully compromises the model, they could potentially control the entire network. To mitigate these risks, decentralized AI models offer a promising solution by distributing the learning process across multiple devices, reducing the impact of a single compromised entity. While this approach alleviates

the communication overhead of transmitting training data, it also introduces new challenges, such as managing model updates across nodes, efficiently aggregating updates, and addressing synchronization issues. These challenges must be carefully considered to develop resilient and secure learning frameworks for CR tasks.

Appendix A

List of Publications

Journal Articles

1. N. Abdel Khalek, D. H. Tashman, and W. Hamouda, "Advances in Machine Learning-Driven Cognitive Radio for Wireless Networks: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 26, no. 2, pp. 1201–1237, 2024, doi: 10.1109/COMST.2023.3345796.

2. N. Abdel Khalek, N. Abdolkhani and W. Hamouda, "Deep Reinforcement Learning for Joint Power Control and Access Coordination in Energy Harvesting CIoT," in *IEEE Internet of Things Journal*, vol. 11, no. 19, pp. 30833-30846, 1 Oct.1, 2024, doi: 10.1109/JIOT.2024.3416371.

3. N. Abdolkhani, N. Abdel Khalek and W. Hamouda, "Deep Reinforcement Learning for EH-Enabled Cognitive-IoT Under Jamming Attacks," in *IEEE Internet of Things Journal*, vol. 11, no. 24, pp. 40800-40813, 15 Dec.15, 2024, doi: 10.1109/JIOT.2024.3457012.

4. N. Abdolkhani, N. Abdel Khalek, W. Hamouda and I. Dayoub, "Deep Reinforcement Learning for Joint Time and Power Management in SWIPT-EH CIoT," in *IEEE Communications Letters*, vol. 29, no. 4, pp. 660-664, April 2025, doi: 10.1109/LCOMM.2025.3536182.

5. N. Abdel Khalek and W. Hamouda, "Improving Secrecy Capacity in the Face of Eavesdropping with Actor-Critic Deep Reinforcement Learning," in *IEEE Open Journal of the Communications Society*, 2025 (Under Review).

Conference Proceedings

1. N. Abdel Khalek and W. Hamouda, ”Unsupervised Two-Stage Learning Framework for Cooperative Spectrum Sensing,” in *Proc. IEEE International Conference on Communications (ICC)*, Montréal, QC, Canada, 2021, pp. 1-6, doi: 10.1109/ICC42927.2021.9500681.

2. N. Abdel Khalek and W. Hamouda, ”Intelligent Spectrum Sensing: An Unsupervised Learning Approach Based on Dimensionality Reduction,” in *Proc. IEEE International Conference on Communications (ICC)*, 2022, pp. 171-176, doi: 10.1109/ICC45855.2022.9839170.

3. N. Abdel Khalek and W. Hamouda, ”DeepSense: An Unsupervised Deep Clustering Approach for Cooperative Spectrum Sensing,” in *Proc. IEEE International Conference on Communications (ICC)*, Rome, Italy, 2023, pp. 1868-1873, doi: 10.1109/ICC45041.2023.10279156.

4. N. Abdel Khalek and W. Hamouda, ”DEAP Learning: A Data-Driven Approach to Unsupervised Cooperative Spectrum Sensing,” in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Kuala Lumpur, Malaysia, 2023, pp. 6389-6394, doi: 10.1109/GLOBECOM54140.2023.10437464. **(BEST PAPER AWARD)**.

5. N. Abdel Khalek and W. Hamouda, ”G-VAP: A Generative Variational Autoencoder Approach for Enhanced Cooperative Sensing,” in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Cape Town, South Africa, 2024, pp. 1245-1250, doi: 10.1109/GLOBECOM52923.2024.10901477.

6. N. Abdel Khalek and W. Hamouda, ”Optimizing Spectrum Efficiency in Hybrid Cognitive Radios Through Unsupervised Learning,” in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Cape Town, South Africa, 2024, pp. 1251-1256, doi: 10.1109/GLOBECOM52923.2024.10901689.

7. N. Abdel Khalek and W. Hamouda, “Deep Federated Representations for Distributed and Secure Spectrum Sensing in Large-Scale CRNs” in *Proc. IEEE International Communications Conference (ICC)*, Montréal, QC, Canada, 2025 (To Appear).

Bibliography

[1] S. Sinha, "State of IoT 2024: Number of connected IoT devices growing 13% to 18.8 billion globally," 2024. [Online]. Available: https://iot-analytics.com/number-connected-iot-devices/

[2] Ericsson, "Technology Trends - Cognitive Networks," Mar 2022. [Online]. Available: https://youtu.be/iWas4WXdFeI

[3] Wu, Cheng and Wang, Yiming and Yin, Zhijie, "Realizing Railway Cognitive Radio: A Reinforcement Base-Station Multi-Agent Model," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1452–1467, 2019.

[4] R. Pal, N. Gupta, A. Prakash, R. Tripathi, and J. J. Rodrigues, "Deep reinforcement learning based optimal channel selection for cognitive radio vehicular ad-hoc network," *IET Commun.*, vol. 14, no. 19, pp. 3464–3471, 2020.

[5] X. Liu, C. Sun, M. Zhou, B. Lin, and Y. Lim, "Reinforcement learning based dynamic spectrum access in cognitive Internet of Vehicles," *China Commun.*, vol. 18, no. 7, pp. 58–68, 2021.

[6] H. Xie, R. Lin, J. Wang, M. Zhang, and C. Cheng, "Power Allocation of Energy Harvesting Cognitive Radio Based on Deep Reinforcement Learning," in *Proc. Int. Conf. Commun. Informat. Syst. (ICCIS)*, 2021, pp. 45–49.

[7] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial Neural Networks-Based Machine Learning for Wireless Networks: A Tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 2019.

[8] N. A. Khalek, and W. Hamouda, "Unsupervised Two-Stage Learning Framework for Cooperative Spectrum Sensing," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, 2021, pp. 1–6.

[9] N. A. Khalek and W. Hamouda, "Intelligent Spectrum Sensing: An Unsupervised Learning Approach Based on Dimensionality Reduction," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2022, pp. 171–176.

[10] N. A. Khalek, and W. Hamouda, "DeepSense: An Unsupervised Deep Clustering Approach for Cooperative Spectrum Sensing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2023, pp. 1868–1873.

[11] N. A. Khalek and W. Hamouda, "DEAP Learning: A Data-Driven Approach to Unsupervised Cooperative Spectrum Sensing," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2023, pp. 6389–6394.

[12] N. A. Khalek, and W. Hamouda, "G-VAP: A Generative Variational Autoencoder Approach for Enhanced Cooperative Sensing," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2024, pp. 1245–1250.

[13] N. A. Khalek and W. Hamouda, "Optimizing Spectrum Efficiency in Hybrid Cognitive Radios Through Unsupervised Learning," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2024, pp. 1251–1256.

[14] N. A. Khalek, N. Abdolkhani, and W. Hamouda, "Deep Reinforcement Learning for Joint Power Control and Access Coordination in Energy Harvesting CIoT," *IEEE Internet Things J.*, vol. 11, no. 19, pp. 30 833–30 846, 2024.

[15] N. Abdolkhani, N. A. Khalek, and W. Hamouda, "Deep Reinforcement Learning for EH-Enabled Cognitive-IoT Under Jamming Attacks," *IEEE Internet Things J.*, vol. 11, no. 24, pp. 40 800–40 813, 2024.

[16] N. Abdolkhani, N. A. Khalek, W. Hamouda, and I. Dayoub, "Deep Reinforcement Learning for Joint Time and Power Management in SWIPT-EH CIoT," *IEEE Commun. Lett.*, vol. 29, no. 4, pp. 660–664, 2025.

[17] N. A. Khalek, and W. Hamouda, "Deep Federated Representations for Distributed and Secure Spectrum Sensing in Large-Scale CRNs," in *Proc. IEEE Int. Commun. Conf. (ICC)*, 2025, pp. 1–6.

[18] N. A. Khalek, D. H. Tashman, and W. Hamouda, "Advances in Machine Learning-Driven Cognitive Radio for Wireless Networks: A Survey," *IEEE Commun. Surv. Tutor.*, vol. 26, no. 2, pp. 1201–1237, 2024.

[19] M. El Tanab and W. Hamouda, "Resource Allocation for Underlay Cognitive Radio Networks: A Survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1249–1276, 2017.

[20] J. M. Moualeu, W. Hamouda, and F. Takawira, "Cognitive Coded Cooperation in Underlay Spectrum-Sharing Networks Under Interference Power Constraints," *IEEE Trans. Veh. Tech.*, vol. 66, no. 3, pp. 2099–2113, 2017.

[21] A. Ali and W. Hamouda, "Power-Efficient Wideband Spectrum Sensing for Cognitive Radio Systems," *IEEE Trans. Veh. Tech.*, vol. 67, no. 4, pp. 3269–3283, 2018.

[22] A. Ali and W. Hamouda, "A Novel Spectrum Monitoring Algorithm for OFDM-Based Cognitive Radio Networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2015, pp. 1–6.

[23] A. Ali and W. Hamouda, "Advances on Spectrum Sensing for Cognitive Radio Networks: Theory and Applications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1277–1304, 2017.

[24] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A Survey on Machine-Learning Techniques in Cognitive Radios," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1136–1159, 2013.

[25] A. Krayani, M. Baydoun, L. Marcenaro, Y. Gao, and C. S. Regazzoni, "Smart Jammer Detection for Self-Aware Cognitive UAV Radios," in *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, 2020, pp. 1–7.

[26] A. Krayani, M. Baydoun, L. Marcenaro, A. S. Alam, and C. Regazzoni, "Self-Learning Bayesian Generative Models for Jammer Detection in Cognitive-UAV-Radios," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2020, pp. 1–7.

[27] M. S. Abdalzaher, M. Elwekeil, T. Wang, and S. Zhang, "A Deep Autoencoder Trust Model for Mitigating Jamming Attack in IoT Assisted by Cognitive Radio," *IEEE Syst. J.*, pp. 1–11, 2021.

[28] N. Liu, X. Tang, D. Xu, D. Wang, D. Zhai, and R. Zhang, "A Learning Approach Towards Secure Cognitive Networks with UAV Relaying and Active Jamming," in *Proc. Int. Conf. on Wireless Commun. and Signal Proc. (WCSP)*, 2021, pp. 1–6.

[29] X. Li, S. Cheng, N. Zhao, and N. Yao, "A Joint Strategy for CUAV-based Traffic Offloading via Deep Reinforcement Learning," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2021, pp. 01–06.

[30] M. C. Hlophe and B. T. Maharaj, "QoS provisioning and energy saving scheme for distributed cognitive radio networks using deep learning," *J. of Commun. and Netw.*, vol. 22, no. 3, pp. 185–204, 2020.

[31] Y. Fan, W. Xu, C.-H. Lee, S. Wu, F. Yang, and P. Zhang, "Machine Learning-Based Energy-Spectrum Two-Dimensional Cognition in Energy Harvesting CRNs," *IEEE Access*, vol. 8, pp. 158 911–158 927, 2020.

[32] A. Paul and S. P. Maity, "Machine Learning for Spectrum Information and Routing in Multihop Green Cognitive Radio Networks," *IEEE Trans. on Green Commun. and Netw.*, vol. 6, no. 2, pp. 825–835, 2022.

[33] T.-D. Le and G. Kaddoum, "LSTM-Based Channel Access Scheme for Vehicles in Cognitive Vehicular Networks With Multi-Agent Settings," *IEEE Trans. on Veh. Technol.*, vol. 70, no. 9, pp. 9132–9143, 2021.

[34] M. A. Hossain, R. M. Noor, S. R. Azzuhri, M. R. Z'aba, I. Ahmedy, S. S. Anjum, W. M. Shah, and K.-L. A. Yau, "Faster convergence of Q-learning in cognitive radio-VANET scenario," in *Advances in Electronics Engineering*. Springer, 2020, pp. 171–181.

[35] M. Farrukh, A. Krayani, M. Baydoun, L. Marcenaro, Y. Gao, and C. S. Regazzoni, "Learning a Switching Bayesian Model for Jammer Detection in the Cognitive-Radio-Based Internet of Things," in *Proc. IEEE World Forum on Internet of Things (WF-IoT)*, 2019, pp. 380–385.

[36] S. Guo and X. Zhao, "Deep Reinforcement Learning Optimal Transmission Algorithm for Cognitive Internet of Things With RF Energy Harvesting," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 1216–1227, 2022.

[37] K.-L. A. Yau, P. Komisarczuk, and P. D. Teal, "Performance Analysis of Reinforcement Learning for Achieving Context Awareness and Intelligence in Mobile Cognitive Radio Networks," in *Proc. IEEE Int. Conf. on Adv. Info. Netw. and Appl.*, 2011, pp. 1–8.

[38] P. Ghasemzadeh, M. Hempel, and H. Sharif, "GS-QRNN: A High-Efficiency Automatic Modulation Classifier for Cognitive Radio IoT," *IEEE Internet of Things J.*, vol. 9, no. 12, pp. 9467–9477, 2022.

[39] A. Shamsoshoara, M. Khaledi, F. Afghah, A. Razi, J. Ashdown, and K. Turck, "A Solution for Dynamic Spectrum Management in Mission-Critical UAV Networks," in *Proc. IEEE Int. Conf. on Sensing, Commun., and Netw. (SECON)*, 2019, pp. 1–6.

[40] A. Shamsoshoara, M. Khaledi, F. Afghah, A. Razi, and J. Ashdown, "Distributed Cooperative Spectrum Sharing in UAV Networks Using Multi-Agent Reinforcement Learning," in *Proc. IEEE Consumer Commun. & Netw. Conf. (CCNC)*, 2019, pp. 1–6.

[41] N. Abdel Khalek and W. Hamouda, "Learning-Based Cooperative Spectrum Sensing in Hybrid Underlay-Interweave Secondary Networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2020, pp. 1–6.

[42] S. Hu, X. Chen, W. Ni, E. Hossain, and X. Wang, "Distributed Machine Learning for Wireless Communication Networks: Techniques, Architectures, and Applications," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1458–1493, 2021.

[43] Z. Chen, Y.-Q. Xu, H. Wang, and D. Guo, "Federated Learning-Based Cooperative Spectrum Sensing in Cognitive Radio," *IEEE Commun. Lett.*, vol. 26, no. 2, p. 330–334, Feb. 2022.

[44] Y. Zhang, Q. Wu, and M. Shikh-Bahaei, "Vertical Federated Learning Based Privacy-Preserving Cooperative Sensing in Cognitive Radio Networks," in *IEEE Globecom Workshops (GC Wkshps)*, 2020, pp. 1–6.

[45] M. C. Hlophe, B. T. Maharaj, and M. M. Sande, "Energy-Efficient Transmissions in Federated Learning-Assisted Cognitive Radio Networks," in *Proc. IEEE Int. Conf. on Commun. Technol. (ICCT)*, 2021, pp. 216–222.

[46] Z. Gao, A. Li, Y. Gao, B. Li, Y. Wang, and Y. Chen, "FedSwap: A Federated Learning based 5G Decentralized Dynamic Spectrum Access System," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des. (ICCAD)*, 2021, pp. 1–6.

[47] T. D. Ponnimbaduge Perera, D. N. K. Jayakody, S. K. Sharma, S. Chatzinotas, and J. Li, "Simultaneous Wireless Information and Power Transfer (SWIPT): Recent Advances and Future Challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 264–302, 2018.

[48] Y. Alsaba, S. K. A. Rahim, and C. Y. Leow, "Beamforming in Wireless Energy Harvesting Communications Systems: A Survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1329–1360, 2018.

[49] F. Zeng and X. Liu, "Performance Analysis of Adaptive Sensing in Cognitive Radio Networks with Hybrid Interweave-Underlay," in *Proc. Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Jul. 2017, p. 1576–1581.

[50] F. Azmat, Y. Chen, and N. Stocks, "Analysis of Spectrum Occupancy Using Machine Learning Algorithms," *IEEE Trans. on Veh. Tech.*, vol. 65, no. 9, pp. 6853–6860, Sep. 2016.

[51] C. H. A. Tavares and T. Abrão, "Bayesian estimators for cooperative spectrum sensing in cognitive radio networks," in *Proc. IEEE URUCON*, Oct 2017, pp. 1–4.

[52] J. Perez, I. Santamaria, and J. Via, "Adaptive EM-Based Algorithm for Cooperative Spectrum Sensing in Mobile Environments," in *IEEE Stat. Signal Process. Worksh. (SSP)*, June 2018, pp. 732–736.

[53] J. Xie, J. Fang, C. Liu, and L. Yang, "Unsupervised Deep Spectrum Sensing: A Variational Auto-Encoder Based Approach," *IEEE Trans. Veh. Tech.*, vol. 69, no. 5, pp. 5307–5319, 2020.

[54] B. Soni, D. K. Patel, and M. Lopez-Benitez, "Long Short-Term Memory Based Spectrum Sensing Scheme for Cognitive Radio using Primary Activity Statistics," *IEEE Access*, vol. 8, pp. 97 437–97 451, 2020.

[55] D. Li, D. Zhang, and J. Cheng, "A Novel Polarization Enabled Full-Duplex Hybrid Spectrum Sharing Scheme For Cognitive Radios," *IEEE Commun. Lett.*, vol. 23, no. 3, p. 530–533, Mar. 2019.

[56] M. R. Vyas, D. K. Patel, and M. Lopez-Benitez, "Artificial Neural Network Based Hybrid Spectrum Sensing Scheme for Cognitive Radio," in *Proc. IEEE Int. Symp. Personal, Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, p. 1–7.

[57] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, "Machine Learning Techniques for Cooperative Spectrum Sensing in Cognitive Radio Networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 11, pp. 2209–2221, November 2013.

[58] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[59] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[60] N. Abdel Khalek and W. Hamouda, "From Cognitive to Intelligent Secondary Cooperative Networks for the Future Internet: Design, Advances, and Challenges," *IEEE Netw.*, pp. 1–8, 2020.

[61] I. Jolliffe, "Principal Component Analysis," *Encyclopedia of statistics in behavioral science*, 2005.

[62] X. Li, J. Fang, W. Cheng, H. Duan, Z. Chen, and H. Li, "Intelligent Power Control for Spectrum Sharing in Cognitive Radios: A Deep Reinforcement Learning Approach," *IEEE Access*, vol. 6, pp. 25 463–25 473, 2018.

[63] M. R. Vyas, D. K. Patel, and M. Lopez-Benitez, "Artificial Neural Network Based Hybrid Spectrum Sensing Scheme for Cognitive Radio," in *Proc. IEEE Int. Symp. Personal, Indoor, Mobile Radio Commun. (PIMRC)*, 2017, pp. 1–7.

[64] Z. Shi, W. Gao, S. Zhang, J. Liu, and N. Kato, "Machine Learning-Enabled Cooperative Spectrum Sensing for Non-Orthogonal Multiple Access," *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 5692–5702, 2020.

[65] S. Zhang, Y. Wang, Y. Zhang, P. Wan, and J. Zhuang, "A Novel Clustering Algorithm Based on Information Geometry for Cooperative Spectrum Sensing," *IEEE Syst. J.*, vol. 15, no. 2, pp. 3121–3130, 2021.

[66] A. Subekti, H. F. Pardede, R. Sustika, and Suyoto, "Spectrum Sensing for Cognitive Radio using Deep Autoencoder Neural Network and SVM," in *Int. Conf. on Radar, Antenna, Microwave, Electronics, and Telecomm.*, 2018, pp. 81–85.

[67] Q. Cheng, Z. Shi, D. N. Nguyen, and E. Dutkiewicz, "Sensing OFDM Signal: A Deep Learning Approach," *IEEE Trans. on Commun.*, vol. 67, no. 11, pp. 7785–7798, 2019.

[68] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[69] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner, "β-VAE: Learning basic visual concepts with a constrained variational framework." *ICLR (Poster)*, vol. 3, 2017.

[70] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.

[71] G. Xu, Y. Wang, B. Zheng, and J. Li, "Deep Semi-Supervised Learning-based Spectrum Sensing at Low SNR," *IEEE Commun. Lett.*, p. 1–1, 2024.

[72] Y. Li, G. Lu, Z. Li, and Y. Ye, "Graph Neural Network Based Cooperative Spectrum Sensing for Cognitive Radio," in *Proc. IEEE Wirel. Commun. and Netw. Conf. (WCNC)*, 2024, pp. 1–6.

[73] J. H. Bae and M. Kim, "Performance Improvement of Cooperative Spectrum Sensing Based on Dequantization Neural Networks," *IEEE Wirel. Commun. Lett.*, vol. 13, no. 5, p. 1354–1358, May 2024.

[74] W. Li, G. Chen, X. Zhang, N. Wang, D. Ouyang, and C. Chen, "Efficient and Secure Aggregation Framework for Federated-Learning-Based Spectrum Sharing," *IEEE Internet Things J.*, vol. 11, no. 10, p. 17223–17236, May 2024.

[75] Y. Xu, P. Cheng, Z. Chen, Y. Li, and B. Vucetic, "Mobile Collaborative Spectrum Sensing for Heterogeneous Networks: A Bayesian Machine Learning Approach," *IEEE Trans. Signal Process.*, vol. 66, no. 21, p. 5634–5647, Nov. 2018.

[76] B. Hamdaoui, B. Khalfi, and N. Zorba, "Dynamic Spectrum Sharing in the Age of Millimeter-Wave Spectrum Access," *IEEE Netw.*, vol. 34, no. 5, pp. 164–170, 2020.

[77] F. Li, K.-Y. Lam, X. Li, Z. Sheng, J. Hua, and L. Wang, "Advances and Emerging Challenges in Cognitive Internet-of-Things," *IEEE Trans. Ind. Inform.*, vol. 16, no. 8, pp. 5489–5496, 2020.

[78] X. Tan, L. Zhou, H. Wang, Y. Sun, H. Zhao, B.-C. Seet, J. Wei, and V. C. M. Leung, "Cooperative Multi-Agent Reinforcement-Learning-Based Distributed Dynamic Spectrum Access in Cognitive Radio Networks," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 19 477–19 488, 2022.

[79] H. Yang, W.-D. Zhong, C. Chen, A. Alphones, and X. Xie, "Deep-Reinforcement-Learning-Based Energy-Efficient Resource Management for Social and Cognitive Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5677–5689, 2020.

[80] M. Lu, B. Zhou, Z. Bu, and Y. Zhao, "A Learning Approach Towards Power Control in Full-Duplex Underlay Cognitive Radio Networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2022, pp. 2017–2022.

[81] W. Lee and K. Lee, "Deep Learning-Aided Distributed Transmit Power Control for Underlay Cognitive Radio Network," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3990–3994, 2021.

[82] A. Alsharoa, N. M. Neihart, S. W. Kim, and A. E. Kamal, "Multi-Band RF Energy and Spectrum Harvesting in Cognitive Radio Networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–6.

[83] X. Chen, X. Xie, Z. Shi, and Z. Fan, "Dynamic Spectrum Access Scheme of Joint Power Control in Underlay Mode Based on Deep Reinforcement Learning," in *Proc. IEEE Int. Conf. Commun. China (ICCC)*, 2020, pp. 536–541.

[84] M. Chu, X. Liao, H. Li, and S. Cui, "Power Control in Energy Harvesting Multiple Access System With Reinforcement Learning," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9175–9186, 2019.

[85] F. Shah-Mohammadi and A. Kwasinski, "Deep Reinforcement Learning Approach to QoE-Driven Resource Allocation for Spectrum Underlay in Cognitive Radio Networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2018, pp. 1–6.

[86] Z. Shi, X. Xie, H. Lu, H. Yang, J. Cai, and Z. Ding, "Deep Reinforcement Learning-Based Multidimensional Resource Management for Energy Harvesting Cognitive NOMA Communications," *IEEE Trans. Commun.*, vol. 70, no. 5, pp. 3110–3125, 2022.

[87] D. H. Tashman, S. Cherkaoui, and W. Hamouda, "Performance Optimization of Energy-Harvesting Underlay Cognitive Radio Networks Using Reinforcement Learning," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, 2023, pp. 1160–1165.

[88] X. Li, J. Fang, W. Cheng, H. Duan, Z. Chen, and H. Li, "Intelligent Power Control for Spectrum Sharing in Cognitive Radios: A Deep Reinforcement Learning Approach," *IEEE Access*, vol. 6, pp. 25 463–25 473, 2018.

[89] A. T. Z. Kasgari, B. Maham, H. Kebriaei, and W. Saad, "Dynamic Learning for Distributed Power Control in Underlaid Cognitive Radio Networks," in *Proc. Int. Wireless Commun. & Mobile Comput. Conf. (IWCMC)*, 2018, pp. 213–218.

[90] H. Zhang, N. Yang, W. Huangfu, K. Long, and V. C. M. Leung, "Power Control Based on Deep Reinforcement Learning for Spectrum Sharing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4209–4219, 2020.

[91] I. AlQerm and B. Shihada, "Enhanced Online Q-Learning Scheme for Energy Efficient Power Allocation in Cognitive Radio Networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2019, pp. 1–6.

[92] T. T. Anh, N. C. Luong, D. Niyato, Y.-C. Liang, and D. I. Kim, "Deep Reinforcement Learning for Time Scheduling in RF-Powered Backscatter Cognitive Radio Networks," in *Proc. IEEE Wireless Commu. Netw. Conf. (WCNC)*, 2019, pp. 1–7.

[93] R. Zhang, X. Li, and N. Zhao, "When DSA Meets SWIPT: A Joint Power Allocation and Time Splitting Scheme Based on Multi-Agent Deep Reinforcement Learning," *IEEE Trans. Veh. Technol.*, vol. 72, no. 2, pp. 2740–2744, 2023.

[94] A. Omidkar, A. Khalili, H. H. Nguyen, and H. Shafiei, "Reinforcement-Learning-Based Resource Allocation for Energy-Harvesting-Aided D2D Communications in IoT Networks," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16 521–16 531, 2022.

[95] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *Proc. IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2015, pp. 1026–1034.

[96] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[97] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," in *Proc. Int. Conf. Mach. Learn. (ICML)*. PMLR, 2014, pp. 387–395.

[98] A. Anzaldo and A. G. Andrade, "Experience Replay-Based Power Control for Sum-Rate Maximization in Multi-Cell Networks," *IEEE Wirel. Commun. Lett.*, vol. 11, no. 11, pp. 2350–2354, 2022.

[99] A. Hossein Zarif, P. Azmi, N. M. Yamchi, M. R. Javana, and E. A. Jorswieck, "AoI Minimization in Energy Harvesting and Spectrum Sharing Enabled 6G Networks," *IEEE Trans. Green Commun. and Netw.*, vol. 6, no. 4, pp. 2043–2054, 2022.

[100] R. Lin, H. Qiu, J. Wang, Z. Zhang, L. Wu, and F. Shu, "Physical-Layer Security Enhancement in Energy-Harvesting-Based Cognitive Internet of Things: A GAN-Powered Deep Reinforcement Learning Approach," *IEEE Internet Things J.*, vol. 11, no. 3, pp. 4899–4913, 2024.

[101] H. Pirayesh and H. Zeng, "Jamming Attacks and Anti-Jamming Strategies in Wireless Networks: A Comprehensive Survey," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 767–809, 2022.

[102] J. Xu, H. Lou, W. Zhang, and G. Sang, "An Intelligent Anti-Jamming Scheme for Cognitive Radio Based on Deep Reinforcement Learning," *IEEE Access*, vol. 8, pp. 202 563–202 572, 2020.

[103] L. Jia, Y. Xu, Y. Sun, S. Feng, and A. Anpalagan, "Stackelberg Game Approaches for Anti-Jamming Defence in Wireless Networks," *IEEE Wireless Commun.*, vol. 25, no. 6, pp. 120–128, 2018.

[104] A. Gouissem, K. Abualsaud, E. Yaacoub, T. Khattab, and M. Guizani, "Game Theory for Anti-Jamming Strategy in Multichannel Slow Fading IoT Networks," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 16 880–16 893, 2021.

[105] Y. Xu, Y. Xu, X. Dong, G. Ren, J. Chen, X. Wang, L. Jia, and L. Ruan, "Convert Harm Into Benefit: A Coordination-Learning Based Dynamic Spectrum Anti-Jamming Approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13 018–13 032, 2020.

[106] Y. Li, S. Bai, and Z. Gao, "A Multi-Domain Anti-Jamming Strategy Using Stackelberg Game in Wireless Relay Networks," *IEEE Access*, vol. 8, pp. 173 609–173 617, 2020.

[107] I. K. Ahmed and A. O. Fapojuwo, "Stackelberg Equilibria of an Anti-Jamming Game in Cooperative Cognitive Radio Networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 1, pp. 121–134, 2018.

[108] X. Wang, J. Wang, Y. Xu, J. Chen, L. Jia, X. Liu, and Y. Yang, "Dynamic Spectrum Anti-Jamming Communications: Challenges and Opportunities," *IEEE Commun. Mag.*, vol. 58, no. 2, pp. 79–85, Feb. 2020.

[109] A. S. Ali, S. Naser, and S. Muhaidat, "Defeating Proactive Jammers Using Deep Reinforcement Learning for Resource-Constrained IoT Networks," in *Proc. IEEE Int. Symp. Pers. Indoor Mob. Radio Commun. (PIMRC)*, 2023, pp. 1–6.

[110] W. Shen, W. Wang, H. Jin, and W. Zhang, "Defend Against Jamming Attacks Using Deep Reinforcement Learning," in *Proc. Int. Symp. Antennas Propag. EM Theory (ISAPE)*, Dec. 2021, pp. 1–3.

[111] M. A. Aref and S. K. Jayaweera, "Robust Deep Reinforcement Learning for Interference Avoidance in Wideband Spectrum," in *Proc. IEEE Cogn. Commun. Aerosp. Appl. Workshop (CCAAW)*, Jun. 2019, pp. 1–5.

[112] H. Han, Y. Xu, Z. Jin, W. Li, X. Chen, G. Fang, and Y. Xu, "Primary-User-Friendly Dynamic Spectrum Anti-Jamming Access: A GAN-Enhanced Deep Reinforcement Learning Approach," *IEEE Wireless Commun. Lett.*, vol. 11, no. 2, pp. 258–262, Feb. 2022.

[113] Q. Zhou, Y. Li, Y. Niu, Z. Qin, L. Zhao, and J. Wang, "One Plus One is Greater Than Two: Defeating Intelligent Dynamic Jamming with Collaborative Multi-agent Reinforcement Learning," in *Proc. IEEE Int. Conf. Comput. Commun. (ICCC)*, Dec. 2020, pp. 1522–1526.

[114] Y. Bi, Y. Wu, and C. Hua, "Deep Reinforcement Learning Based Multi-User Anti-Jamming Strategy," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[115] S. Geng, P. Li, X. Yin, H. Lu, R. Zhu, W. Cao, and J. Nie, "The Study on Anti-Jamming Power Control Strategy based on Q-learning," in *Proc. Int. Conf. Intell. Comput. Signal Proc. (ICSP)*, 2022, pp. 182–185.

[116] Y. Chen, Y. Li, D. Xu, and L. Xiao, "DQN-Based Power Control for IoT Transmission against Jamming," in *Proc. IEEE Veh. Technol. Conf. (VTC Spring)*, Jun. 2018, pp. 1–5.

[117] P. K. H. Nguyen, V. H. Nguyen, and V. L. Do, "A Deep Double-Q Learning-based Scheme for Anti-Jamming Communications," in *Proc. Eur. Signal Proc. Conf. (EUSIPCO)*, Jan. 2021, pp. 1566–1570.

[118] A. S. Ali, W. T. Lunardi, L. Bariah, M. Baddeley, M. A. Lopez, J.-P. Giacalone, and S. Muhaidat, "Deep Reinforcement Learning Based Anti-Jamming Using Clear Channel Assessment Information in a Cognitive Radio Environment," in *Proc. Int. Conf. Adv. Commun. Technol. Netw. (CommNet)*, Dec. 2022, pp. 1–6.

[119] M. A. Aref and S. K. Jayaweera, "Spectrum-Agile Cognitive Radios Using Multi-Task Transfer Deep Reinforcement Learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6729–6742, Oct. 2021.

[120] R. Lin, H. Qiu, W. Jiang, Z. Jiang, Z. Li, and J. Wang, "Deep Reinforcement Learning for Physical Layer Security Enhancement in Energy Harvesting Based Cognitive Radio Networks," *Sensors*, vol. 23, no. 2, 2023.

[121] K. Ibrahim, S. X. Ng, I. M. Qureshi, A. N. Malik, and S. Muhaidat, "Anti-Jamming Game to Combat Intelligent Jamming for Cognitive Radio Networks," *IEEE Access*, vol. 9, pp. 137 941–137 956, 2021.

[122] Q. Zhou, Y. Li, and Y. Niu, "A Countermeasure Against Random Pulse Jamming in Time Domain Based on Reinforcement Learning," *IEEE Access*, vol. 8, pp. 97 164–97 174, 2020.

[123] R. S. Sutton and A. Barto, *Reinforcement learning: An introduction*. The MIT Press, 2020.