ARTIFICIAL INTELLIGENCE-BASED RADIO RESOURCE MANAGEMENT FOR WIRELESS NETWORKS

DISSERTATION

for the Degree of

DOCTOR OF PHILOSOPHY
(Electrical Engineering)

DO VINH QUANG

MAY 2020

Artificial Intelligence-based Radio Resource Management for Wireless Networks

DISSERTATION

Submitted in Partial Fulfillment
of the Requirements for the
Degree of

DOCTOR OF PHILOSOPHY
(Electrical Engineering)

at the

UNIVERSITY OF ULSAN

by

Do Vinh Quang
May 2020

Supervisor: Professor In-Soo Koo

Publication No. _____

Artificial Intelligence-based Radio Resource Management for Wireless Networks

Approved by Supervisory Committee:

Prof. Hyung-Yun Kong, Chair

Prof. In-Soo Koo, Supervisor

Prof. Vladimir V. Shakhov

Prof. Myung-Kyun Kim

Prof. Young-Tae Noh

Department of Electrical and Computer Engineering
University of Ulsan, South Korea
Date: May, 2020

VITA

Do Vinh Quang was born in Tay Ninh City, Vietnam, in 1986. He received his bachelor's degree in Electrical and Electronics Engineering from Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2009, and his master's degree in Electronics and Computer Science from Royal Melbourne Institute of Technology (RMIT), Melbourne, Australia, in 2012. Since March 2014, he has been working as a lecturer at the Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Vietnam.

Since March 2017, he has been pursuing his Ph.D. degree in Electrical and Computer Engineering at the University of Ulsan (UOU), South Korea, under the supervision of Professor Insoo Koo. His current research focuses on artificial intelligence techniques (e.g., reinforcement learning algorithms and deep neural networks) and their applications to dense wireless networks with energy-harvesting base stations.

Dedicated to my grateful family
for their love and support

ACKNOWLEDGMENTS

First and foremost, I would like to express my profound attitude to my advisor, *Professor Insoo Koo*, for offering me the opportunity to be a member of his research group. I am indebted to him for his kindness, constant support, encouragement, and persistent guidance throughout my Ph.D. program. His step-by-step guidance is invaluable since it has helped me a lot in doing research, especially when dealing with problems.

I would also like to thank the members of my Ph.D. supervisory committee for useful comments and for contributing their broad perspective in improving the quality of this dissertation.

I am grateful to other members of the Multimedia Communications System Laboratory, University of Ulsan, for their friendship, enthusiastic help, and cheerfulness during my study in South Korea. More specifically, I would like to thank *Dr. Tran Nhut Khai Hoan* and *Dr. Vu Van Hiep* for their great support and valuable discussion.

I really appreciate the BK21+ program for financial support during my study at the University of Ulsan.

Last but not least, I would like to thank my family for their spiritual support and encouragement, especially my loving wife, *Nguyen Thi Phuong Nghi*, and son, *Do Minh Dang*, who have always supported every endeavor of mine.

Do Vinh Quang

Ulsan, South Korea, May - 2020.

ABSTRACT

Artificial Intelligence-based Radio Resource Management for Wireless Networks

by

Do Vinh Quang

Supervisor: Prof. In-Soo Koo

Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy (Electrical Engineering)

May 2020

Cognitive radio is considered an effective solution to the problem of spectrum scarcity, which allows secondary users (SUs) to opportunistically access licensed spectrum bands that are temporarily unused by the primary users (PUs). Therefore, it has attracted much attention from both academic and industrial communities in recent years. In cognitive radio networks (CRNs), the SUs frequently sense the presence of the PU on the licensed channels and then transmit data on unoccupied channels. In modern communication systems, the security of CRNs is critical, since the legitimate communication in CRNs might be vulnerable to hidden eavesdroppers due to the open characteristics of the networks. Furthermore, energy conservation has been a primary concern for energy-harvesting powered CRNs, in which the SUs harvest energy from ambient sources, such as solar power, wind power, and radio frequency (RF) energy. Each energy-harvesting node uses its limited energy for spectrum sensing, data processing, and data transmission. To improve spectral efficiency and energy efficiency, network operators tend to deploy more and more small-cell radio networks with short-range and low-power base stations (BSs). Such a deployment can enhance network coverage and capacity in highly populated areas. However, it also brings challenges to efficient resource allocation due to the stochastic property of mobile subscribers and the intensive characteristic of dense networks. Hence, how to effectively manage scarce resources, such as spectrum and energy, is of critical importance in the design of energy harvesting-based wireless networks. Future wireless networks will become

more intelligent with the assistance of artificial intelligence (AI) techniques, such as machine learning (ML), optimization theory, game theory, and meta-heuristics. Among them, reinforcement learning (RL) methods and deep neural networks (DNNs), which are two of the most important sub-fields of ML, are well known for their useful applications in wireless networks. Accordingly, RL methods and DNNs have shown their advantages in empowering wireless communication systems in terms of network operation and optimization. Therefore, it is essential to employ these innovative techniques into future mobile networks to ensure long-term and maintenance-free operation of energy harvesting-based networks. In this dissertation, we study the applications of AI techniques for efficient resource management and security improvement in energy harvesting-based wireless networks. We aim to find the optimal resource management scheme that can ensure long-term network performance.

In the first part of this dissertation, we investigate the problem of energy-efficient data communications in an energy-harvesting cognitive radio network, in which SUs harvest energy from solar power and opportunistically access a time-slotted primary channel for data transmission. However, legitimate communication can be vulnerable to external attacks that are carried out by hidden eavesdroppers. Therefore, we propose two energy-efficient data encryption schemes for a SU in CRNs to increase the security level under energy constraints. More specifically, based on the sensing result at the beginning of each time slot, the SU decides whether to stay silent to save energy or to transmit data to the destination. The SU also needs to choose an appropriate private-key data encryption method to maximize data security in the long run. In the first scheme, the information about the environment (e.g., the activity of the PU and the model of harvested energy) is available to the SUs. Hence, we model the problem as the framework of a partially observable Markov decision process (POMDP). We then use a value iteration-based method to solve the formulated problem. In the second scheme, the SU will interact with the environment through a sequential decision process. During this process, the SU can decide its operation mode based on a reinforcement learning-based algorithm, which can maximize its long-term data security.

In the second part of this dissertation, we study an optimal power allocation policy for energy-efficient data transmissions in a wireless sensor network in the presence of a full-duplex (FD) eavesdropper. In this network, a sensor node (i.e., the source) powered by renewable energy wants to transmit data to a cluster head (i.e., the destination). The eavesdropper with FD capability can opportunistically launch jamming attacks to the destination. We aim to find the optimal power allocation scheme for the source to maximize its

long-term secrecy rate. We model the problem of transmit power allocation as the framework of a Markov decision process and investigate the formulated problem in two different scenarios. In the first scenario, we propose a POMDP-based method to solve the problem using value iteration-based dynamic programming with the assumption that the information about the harvested energy and the model of jamming activities of the eavesdropper is available to the system. In the second scenario, we use a learning-based algorithm to help the source find the optimal solution to the power allocation problem through interactions with the environment. We verify the effectiveness of the proposed schemes through numerical simulation results.

The third part of this dissertation mainly presents reinforcement learning-based methods for efficient resource allocation and user scheduling in small-cell networks with energy harvesting. First, we investigate the problem of bandwidth allocation for an operation controller in hierarchical cellular networks consisting of several small-cell base stations (SBSs) that are powered by energy harvesters. We aim to find the optimal bandwidth allocation policy in order to enhance user satisfaction and energy efficiency within the constraints of energy harvesting and bandwidth sharing. However, the arrivals of harvested energy and traffic requests are unknown in advance, so it is necessary to design a learning algorithm for the controller to predict the system dynamics before making decisions about bandwidth allocation. Therefore, we employ a natural actor-critic algorithm to help the controller effectively allocate bandwidth to the SBSs. Then, we introduce an actor-critic deep learning framework for efficient user association and bandwidth allocation in dense mobile networks with green base stations. The agent of the proposed algorithm learns about the evolution of the environment through trial and error experience. In this framework, we use deep neural networks to approximate the policy and the value functions so that the algorithm can work effectively with large-scale problems. Simulation results show that the proposed methods can improve network performance in the long run.

Then, we consider the problem of resource sharing in wireless virtualized networks with energy harvesting, where several virtual network operators (VNOs) lease spectrum resources from a mobile network operator (MNO) to provide data services to their subscribers. We aim to find the optimal spectrum leasing schemes based on deep reinforcement learning (DRL) algorithms in order to help the VNOs provide users with the best performance while ensuring the minimal leasing costs. Since the spectrum resources are limited, the VNOs need to compete for them by announcing their requested spectrum sizes to the

MNO. We investigate the spectrum competition problem in both regular virtualized networks and cognitive virtualized networks with energy-harvesting base stations. In the first scenario, each VNO leases spectrum only through a long-term contract with the MNO. In the second scenario, the VNOs can obtain spectrum resources via both spectrum sensing and leasing contract. We formulate the resource leasing problem in the mentioned scenarios as the framework of a sequential decision-making process. We then develop a DRL algorithm, which is a combination of DNNs and RL, for a VNO to learn the optimal leasing policy by interacting with the environment. We compare the performance of the proposed methods with other traditional learning and non-learning methods.

Finally, we summarize the main contributions of this dissertation and discuss future research directions regarding deep reinforcement learning and its applications in modern wireless networks.

Contents

List of Figures

Nomenclature

Notation Description

AI	Artificial Intelligence
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BS	Base Station
CNN	Convolutional Neural Network
CRN	Cognitive Radio Network
CSI	Channel State Information
CSS	Cooperative Spectrum Sensing
DNN	Deep Neural Network
DRL	Deep Reinforcement Learning
DSA	Dynamic Spectrum Access
FC	Fusion Center
FD	Full Duplex
HD	Half Duplex
ML	Machine Learning
POMDP	Partially Observable Markov Decision Process
PU	Primary User
QoS	Quality of Service
RAN	Radio Access Network
RL	Reinforcement Learning
SBS	Small-cell Base Station
SDN	Software Defined Networking
SINR	Signal-to-Interference-plus-Noise Ratio
SNR	Signal-to-Noise Ratio
SR	Secrecy Rate
SU	Secondary User
UE	User Equipment

Chapter 1

Introduction

1.1 Background

1.1.1 Cognitive Radio Network

Cognitive radio (CR) has been considered an effective solution to the problem of spectrum scarcity and under-utilization in wireless networks [2]. In cognitive radio networks (CRNs), secondary users (SUs) can opportunistically access licensed spectrum bands that are temporarily unoccupied by primary users (PUs) at a particular time and a specific location, which is also known as dynamic spectrum access (DSA) [3]. With DSA, the spectrum holes should be accurately determined by spectrum sensing techniques in order to protect the PUs from potential collisions. A SU can adapt to the operation of a PU on the licensed channel by modifying its radio operating parameters (e.g., transmit power, modulation mode, and carrier frequency) to sense and monitor the presence of the PU on that channel [4]. If the PU is sensed inactive, the SU can utilize the corresponding channel for data transmission without causing harmful interference to the PU. The detection performance can be further improved by using cooperative spectrum sensing (CSS) techniques. The core idea is that several SUs perform spectrum sensing individually and send their local sensing information to a fusion center (FC), where the data will be combined using specific rules to generate a final decision about the state of the PU. The FC then broadcasts the decision to the SUs in the network. In general, CSS can provide the SUs with more accurate decisions (i.e., a higher detection probability and a smaller false-alarm probability). However, the SUs should always be aware of the changes in the radio environment, which might happen

from time to time and affect the final sensing results.

1.1.2 Dense Networks with Energy Harvesting

The ultra-dense networking is considered a promising network architecture for future wireless communication. A typical dense network usually includes low power and small-cell base stations, such as microcell, picocell, and femtocell base stations [5]. Ultra-dense networking technology is expected to enhance the overall performance of the network in terms of network coverage, spectral efficiency, and energy efficiency. In dense networks, the small-cell base stations (SBSs) are usually powered by energy-harvesting modules, which can extract energy from ambient sources, such as solar power, wind power, and radio frequency (RF) energy. The recent advances in energy harvesting technologies have provided the networks with an increased lifetime and ease of maintenance. For example, it is essential to deploy energy harvesting in an area that is not easy to reach by the human, where the supply of traditional grid power is not guaranteed. The harvested energy is usually stored in rechargeable batteries with finite capacity to support the autonomous operation of the network. However, densely deploying SBSs in highly populated areas also brings new challenges to efficient resource utilization and power conservation due to the high density of subscribers and the stochastic property of energy harvesting. Therefore, it is necessary to design an energy-efficient resource management scheme based on artificial intelligence to improve long-term network performance.

1.1.3 Deep Reinforcement Learning

Reinforcement learning [1] is an area of machine learning, which allows a learning agent to learn the optimal decision policy through a decision-making process. In wireless communication, an RL agent periodically selects actions to interact with the network environment and then receives feedback signals that reflect the effectiveness of the actions taken. An RL problem is usually formulated as the framework of a Markov decision process (MDP), which is composed of several components, such as a state space, an action space, a reward function, and a state transition distribution. At each particular time step in the MDP, the agent first observes the environment state and then applies an action to the environment. Based on the received feedback and the state transition, the agent gradually optimizes its strategy to obtain better rewards in the future. Recently, RL has been consid-

ered one of the efficient approaches for resource management in time-variant systems, such as wireless communication networks and cloud computing networks [6]. Furthermore, the emerging deep reinforcement learning is considered an enhanced version of traditional RL, which provides better solutions to large-scale and complicated problems. DRL embraces the advantages of DNNs to improve the learning process and the performance of the RL algorithms. In sophisticated network optimization, DRL can help network operators to solve complex resource management problems (e.g., power allocation and spectrum allocation) to achieve the optimal solutions without complete information about the network entities. DRL also allows the network controller to learn the dynamics of the network environment. Therefore, the controller can efficiently perform network management, such as user association, spectrum assignment, and transmit power allocation in large-scale networks, where there is a large number of mobile devices.

1.2 Thesis Motivation and Objective

The rapid growth in the number of mobile subscribers and multimedia services has led to increasing demands for the radio spectrum recently. However, many studies show that the current spectrum assignment policy was not so efficient since it causes spectrum resources to be under-utilized. As a consequence, Mitola [7] proposed CRN as a promising alternative to the traditional modes of wireless communications. A cognitive radio node can modify its operational parameters to adapt to the changes in the environment by using specific cognitive radio techniques. In CRNs, unlicensed users (i.e., cognitive users or secondary users) can utilize spectrum holes in the spectrum bands, which are temporarily unused by the primary users, at a particular time and a specific location. However, legitimate transmissions in a CRN might be vulnerable to malicious attacks due to its open, sharing, and random access characteristics. For example, data transmissions might be either disrupted by an active jammer or overheard by a hidden eavesdropper. Hence, it is crucial to guarantee the confidentiality and authenticity of the information traveling through the network.

In recent years, network operators tend to deploy wireless mobile networks by using small-cell base stations, which can enhance network coverage and capacity. However, spectrum efficiency in small-cell networks has become a challenging problem due to the intensive characteristics of dense networks and the stochastic property of mobile users.

Energy harvesting technology, which allows the harvesting devices to obtain energy from ambient sources in the environment, is considered a promising solution to energy conservation in dense networks. Base stations in an energy harvesting-based network are usually equipped with energy-harvesting modules that can regularly recharge their finite-capacity batteries. In addition to spectrum efficiency, energy efficiency has also become one of the major concerns for green wireless networks due to the stochastic arrivals of harvested energy. Therefore, it is essential to attain an efficient resource management policy to improve long-term network performance.

Since conventional network architectures might not satisfy bandwidth-intensive and time-intensive data services in mobile communications, software-defined networking (SDN) [8] and wireless network virtualization (WNV) [9] are emerging as the key technologies to enhance the network utility. WNV is a process of abstracting, slicing, and sharing radio resources in a virtualized way. In mobile cellular networks, WNV allows mobile network operators (MVNOs) to share the same network infrastructure (e.g., licensed spectrum and base stations) owned by a mobile network owner, and it can thus provide better resource utilization. However, deploying WNV, in practice, is much more complicated due to the stochastic characteristics of wireless networks (e.g., time-varying wireless channels, signal attenuation, and user mobility). Applying SDN to WNV can help to simplify the network management process, and thus, can improve the overall performance of the whole network in terms of higher data rates and lower operational costs [10]. However, only a few studies consider the problem of spectrum leasing in SDN-based wireless virtualized networks with energy harvesting and cognitive capabilities.

The objective of this dissertation is to solve the mentioned issues by using artificial intelligence-based methods, such as value iteration-based dynamic programming, reinforcement learning, and deep learning. The contributions of this dissertation are summarized as follows:

(i) First, we investigate a security mode decision policy for a CRN, in which cognitive users are solely powered by non-radio frequency energy harvesters.

(ii) Second, we investigate a novel energy-efficient data transmission scheme in CRNs in the presence of full-duplex active eavesdroppers, which aims to maximize secrecy transmission rate.

(iii) Third, we design different learning-based frameworks for efficient resource allocation

and user scheduling in dense networks with small-cell base stations.

(iv) Finally, we study the optimal spectrum leasing strategy based on deep reinforcement learning for virtualized wireless networks, where the shared spectrum resources are limited, and the base stations are powered by solar energy.

1.3 Thesis Outline

The rest of this thesis is organized as follows. Chapters 2 and 3 present secure data transmission schemes for CRNs under energy constraints. Chapter 4 introduces efficient bandwidth-allocation and user-association schemes in dense mobile networks. Chapter 5 studies competitive spectrum leasing strategies in virtualized wireless networks. Chapter 6 investigates a dynamic task association and resource allocation scheme for green edge computing. A brief description of each chapter is given below.

Chapter 2 introduces energy-efficient data encryption schemes for a CRN to increase the security level under energy limitation constraints. In this CRN, the secondary users harvest energy from solar power while opportunistically accessing a licensed channel for data transmission. The network is assumed to operate in a time-slotted manner. At the beginning of each time slot, the SUs perform spectrum sensing individually and send the local decisions about the state of the primary channel to a fusion center (FC). We first develop a new cooperative spectrum sensing method by using convolutional neural networks, which uses historical sensing data for classification problems, to improve detection performance. We then propose two different methods for an SU to decide its operation mode in order to increase security against a hidden eavesdropper. Based on the sensing result, the SU can decide whether to stay silent to save energy, or to transmit data that is encrypted with appropriate encryption methods. The problem is formulated as the framework of a Markov decision process, and it will be solved by using either a value iteration-based dynamic programming method or a transfer learning actor-critic algorithm.

Chapter 3 studies an optimal transmit power decision policy for energy-efficient data transmissions in a wireless sensor network in the presence of a full-duplex (FD) active eavesdropper. In this network, a sensor node (i.e., the source), which is powered by a wireless energy harvester, needs to send information to a cluster head (i.e., the destination). Meanwhile, an eavesdropper with FD capability tries to affect the legitimate transmissions

by launching jamming attacks towards the destination while eavesdropping. We aim to find an optimal power allocation policy for the source in order to maximize the secrecy transmission rate. We study the problem in two different scenarios. First, the legitimate nodes are assumed to have prior information about the system dynamics (e.g., the arrival of harvested energy and the jamming model of the eavesdropper). Hence, the problem can be solved by using a value iterations method. Second, the legitimate nodes do not know the environmental dynamics in advance, so we propose an actor-critic learning framework to find the solution from practical interactions with the environment.

Chapter 4 introduces efficient user-association and bandwidth-allocation schemes based on reinforcement learning and deep learning for downlink data transmission in dense mobile networks. More specifically, several small cells are deployed in a single macrocell and share the same spectrum band with the macrocell. The small-cell base stations are powered solely by solar-energy harvesters. This chapter is divided into two main parts. In the first part, we aim to find the optimal bandwidth allocation policy in order to enhance network performance in terms of user satisfaction and energy efficiency under energy harvesting and bandwidth sharing constraints. Therefore, we employ an actor-critic reinforcement learning algorithm to find the optimal policy under which the network controller can effectively allocate the limited bandwidth to the SBSs for their data transmissions. The second part is an extended version of the first part, in which we consider not only bandwidth allocation but also user association for small-cell networks. We propose an actor-critic deep learning framework, which is a combination of artificial neural networks and reinforcement learning, to maximize long-term network performance while adhering to constraints on harvested energy and spectrum sharing.

Chapter 5 considers the problem of resource sharing in a virtual mobile network with energy-harvesting base stations, where several virtual network operators (VNOs) lease radio resources (i.e., wireless channels) from a mobile network operator (MNO) to provide data services to their subscribers. We consider this problem in two scenarios: (i) the VNOs only acquire spectrum resources from the MNO via long-term leasing contract, (ii) the VNOs obtain spectrum resources from the MNO via both spectrum sensing and leasing. In both cases, the VNOs want to provide their subscribed users with the best performance while ensuring minimal leasing costs. We formulate the problem as a Markov decision process, during which the VNOs compete with each other for the spectrum resources by dynamically announcing their resource requirements to the MNO. We then develop deep

reinforcement learning algorithms for spectrum leasing strategy at a VNO in the network to learn the optimal resource leasing policy by interacting with the network environment. More specifically, we design a transfer deep Q-learning framework for the first case and a double deep Q network for the second case.

Finally, chapter 6 concludes this thesis and provides discussions on future research directions.

Chapter 2

Energy-Efficient Data Protection in Cognitive Radio Networks

2.1 Introduction

Cognitive radio is a promising solution to the problem of spectrum scarcity and under utilization of wireless communications networks. In cognitive radio networks (CRNs), secondary users (SUs) with cognitive capability can utilize the spectrum bands licensed to the primary users (PUs) for data transmission [11]. To achieve this, the SU needs to adapt to the time-slotted operation of the PU on the channel of interest by modifying its radio operating parameters (e.g., transmit power, modulation mode, and carrier frequency) in order to sense the presence of the PU on that channel. When the PU is sensed as inactive in a particular time slot, the SU can use the licensed channel during that time slot to transmit data. We assume that the SU uses its limited energy for spectrum sensing, data encryption, and data transmission.

2.1.1 Motivation

Many studies concerning energy management problems for energy harvesting nodes have been conducted, primarily to maximize a system's throughput [12–16]. For example, Park and Hong [12] examined a decision policy for spectrum sensing in connection with a detection threshold to enhance the average throughput concerning the constraints of energy limitation and communication collision. Pappas *et al.* [14] examined the two-dimensional

maximum stable throughput region for a simple cognitive system comprising two source-destination pairs. Razaque and Elleithy [16] designed an intelligent decision-making (IDM) model for wireless sensor networks, which allows the sensor node to obtain energy from the Sun, and thus preserves its battery energy in an outdoor environment. Liang *et al.* [17] studied the optimal sensing duration to maximize achievable throughput for a secondary network while sufficiently protecting primary users. There is research that analyzes optimal transmission power and density of secondary transmitters to maximize secondary network throughput under the constraints of a given outage-probability [18]. In addition, Rossi *et al.* [19] explored a multiple-input multiple-output (MIMO) technique for collaborative spectrum sensing for the distributed detection framework in cognitive-radio scenarios. Specifically, the authors focused on the reporting channel in a spectrum-sensing context and exploits the results from decision fusion to improve probability of detection.

In addition, cognitive radio networks (CRNs), like any modern communications system, must guarantee the confidentiality, integrity, and authenticity of the data traveling through the network [20]. However, due to its open and random access nature, wireless communications in CRNs is susceptive to security threats targeting the physical or media access control layers (e.g., passive eavesdropping or radio frequency (RF) jamming). For that reason, a remarkable number of contributions focus mainly on security technologies for CRNs [21]. In particular, Wen *et al.* [22] presented physical layer approaches to defend against security threats in CRNs. The authors first introduced a MIMO technique that guarantees a low probability of interception, and that enhances the confidentiality of the network; then, they proposed an identified scheme based on channel responses to defend against primary user-emulation attacks. Ciuonzo et al. [23] studied channel-aware decision fusion rules to classify the presence of a (either distributed or co-located) multi-antenna jamming device in wireless sensor networks.

Moreover, physical layer security in CRNs has been widely studied to secure wireless transmissions, especially in the presence of a hidden eavesdropper [24, 25]. Besides this, keeping the data classified from prying eyes by using encryption techniques is one of the most feasible solutions to maintain security; but, in reality, it is not easy to implement conventional encryption techniques in CRNs, since the networks have constrained resources (e.g., limited energy or memory). As a consequence, encryption techniques such as symmetric and asymmetric key algorithms are not preferred for data protection in CRNs. Nevertheless, in modern CRNs, wireless energy harvesting technology can ensure the en-

ergy autonomy of the network by using a small rechargeable battery integrated with an energy harvester, thus providing the SUs with redundant energy to improve data security. Therefore, protecting data using encryption methods still attracts a lot of interest in the research community [26–28]. To illustrate, Sen [29] identified numerous security threats to cognitive wireless sensor networks and the defense mechanisms against these vulnerabilities by selecting the most appropriate cryptography algorithm for each class of attack.

Kim et al. [30] proposed a threshold-based security control scheme for a wireless sensor network where a sensor node decides its encryption mode based on the energy level in its battery. If the node has more energy than a certain threshold, it encrypts data using asymmetric-key encryption algorithm without considering the effects of the decision on future performance of the network. Motivated by this work, we propose two energy-efficient data protection schemes for a CRN where the SU can determine its operation mode (e.g., stay silent or transmit encrypted data) in the current time slot by estimating the current decision's effect on future time slots. More specifically, the Advanced Encryption Standard (AES) [31] algorithms for the same data block length with three different cipher keys (AES-128, AES-192, AES-256) are used for data protection. The security levels can be measured by the complexity of the cracking method (for example, bruce force attacks [32]), which increases in relation to the length of the cipher key. An SU that has remaining energy greater than specific thresholds can encrypt data using algorithms with longer key lengths to increase data confidentiality. Hence, the SU can encrypt data using an algorithm with larger key sizes to enhance security, and then transmits the encrypted data on an idle licensed channel. Furthermore, the SU determines the encryption key size based on the impact of spectrum sensing error, the energy causality constraint, and the effect of the current decision on future time slots. The proposed schemes aim to find an optimal policy for the data encryption decision to maximize the long-term security level of the system.

2.1.2 Contributions

Our focus in this chapter is to solve the problem of reaching a data encryption decision that aims to maximize the security of data transmissions in CRNs. In the first proposed scheme, the problem is formulated as the framework of a partially observable Markov decision process (POMDP), and is then solved by using value iteration-based dynamic programming in order to find the optimal decision policy. However, this solution

is rarely directly useful in reality. It is akin to an exhaustive search, looking ahead at all possibilities, computing the probabilities of occurrence and their desirability in terms of expected rewards (i.e., security levels) [1]. The solution relies on the assumption that we know in advance the dynamics of the environment (e.g., an arrival of harvested energy), which is rarely true in practice. In the second proposed scheme, we investigate the problem from a different point of view in which the solution does not require prior information about the environment's dynamics. More specifically, we solve the problem by using model-free reinforcement learning [1], namely, an actor–critic algorithm. The main advantage of the actor–critic solution over the POMDP-based approach is that it does not require complex computations or information about the arrival of harvested energy.

We model the arrival of harvested energy and the primary traffic as a Poisson point process and a time-homogeneous discrete Markov process, respectively. At the beginning of each time slot, the SU carries out spectrum sensing to identify whether the primary channel is busy or not; then, it either stays idle or transmits data on the free channel. Accordingly, to increase the chances for the SU to transmit data on the primary channel and to reduce the probability of collision with the primary user, we also propose a new cooperative spectrum sensing technique using a convolutional neural network (CNN) and historical sensing data. With the second scheme, we employ an actor-critic sequential learning model so the SU can interact with the environment during the Markov process to acquire information on the environment's dynamics. Based on this method, the SU can learn the energy harvesting model and the primary traffic variations from the learning practice. Afterwards, it can either stay idle or select an appropriate key length for data encryption (also known as *action*), and then verify the effect of the decision based on the returned rewards. By repeating this kind of action over time, the SU can establish the policy to make determinations in the future. However, it would take time for the actor–critic learning procedure to converge to an optimal policy, especially with the large size of the state space [33]. To deal with such an issue, we employ the idea of transfer learning, which exploits the historical relevance of the harvested energy model and the primary user's activity in order to speed up the learning process of the conventional actor–critic algorithm [34]. We call this method a transfer learning actor–critic (TLAC) algorithm. The main contributions of this work are summarized as follows:

- We first introduce a new energy harvesting model, which is represented by a trans-

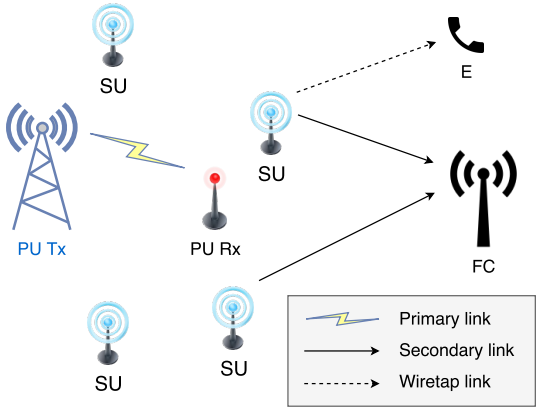

Figure 2.1: A system model of an energy harvesting-based CRN.

formed Poisson distribution proven to give the nearest fit to the empirical measurements of a solar energy harvesting node for time-slotted operation [35].

- We also introduce a new CNN-based technique for cooperative spectrum sensing to enhance the performance of spectrum sensing by increasing the probability of detection while guaranteeing a low probability of false alarm.

- We then formulate the stochastic problem of the data encryption decision policy as the framework of a constrained MDP, and solve the problem by using either the POMDP-based approach or the TLAC algorithm.

The rest of this chapter is organized as follows. In Section 2.2, we introduce the system model of the proposed schemes. In Section 2.3, we describe the new CNN-based cooperative spectrum sensing (CBCSS) technique. In Section 2.4, we present the proposed energy-efficient data protection schemes in CRNs based on POMDP and TLAC. In Section 2.5, we evaluate the performance of the proposed schemes through numerical simulation results. Finally, we conclude this chapter in Section 2.6.

2.2 System Model

We consider a system that consists of a pair of licensed primary users, several secondary transmitters (denoted as SUs), a secondary receiver equipped with a fusion center, and an eavesdropper (E), as shown in Figure 2.1. From now on, we will call the secondary

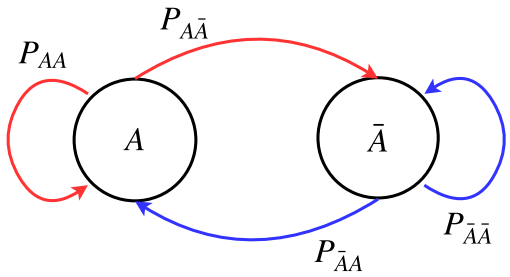

Figure 2.2: Two-state Markov discrete-time model for the primary user's states.

receiver as the fusion center or FC for simplicity. The SUs are assumed to always have data to transmit to the fusion center. Thus, they would try to access the licensed channel of the PUs for data transmission by carrying out cooperative spectrum sensing.

The primary user's states (active $[A]$ and not active $[\bar{A}]$) are assumed to follow a two-state Markov discrete-time process, in which the transition probabilities between the states are denoted $P_{i,j} : i, j \in \{A, \bar{A}\}$, as illustrated in Figure 2.2. The performance of the sensing scheme can be evaluated by using the probability of correct detection P_d and the probability of false alarm P_f. The former represents the probability of detecting the *active* state (A) of the PU accurately, whereas the latter indicates the probability that the PU is identified as active, but it is truly not $(\bar{A})$, each of which are given by

$$P_d = P(H = A|A) \tag{2.1}$$

and

$$P_f = P\left(H = A|\bar{A}\right) \tag{2.2}$$

respectively, where H denotes the state of the primary user as determined by spectrum sensing. Although the PU state transition probabilities are unknown in practical situations, the historical statistics information of the primary channel can be used to estimate the state transition probabilities based on the Markov model [36]. Therefore, we assume that the SU has a prior information about the PU state transition probabilities based on the historical sensing results; and the global information of the network (e.g., channel state information, probabilities of detection and false alarm) are available for all nodes in the network.

The system's operation proceeds as follows. The system is assumed to operate in a time-slotted manner. At the beginning of each time slot, the SUs separately perform spectrum sensing on a selected channel and send the sensing outcomes to the fusion center,

where the data are fused together using a certain rule to make a global decision about the state of the PU on that channel. This global decision is then broadcast to the SUs. If the final sensing result indidates that the PU is inactive, the primary channel is allocated to one of the SUs for data transmission. The SUs take turns using the channel, based on the arrival order of their transmission requests. Each SU can occupy the channel over many time slots until it finishes transmitting data. Meanwhile, the eavesdropper is listening to the communication quietly. Therefore, we are going to investigate a learning framework for cooperative spectrum sensing and energy-efficient data protection schemes against the hidden eavesdropper for the communication between one SU and the fusion center.

We first present a simple but effective cooperative spectrum sensing method based on a CNN to improve the sensing performance. The CNN is constructed and trained to predict the PU states by using individual sensing data as inputs, which leads to specific target outputs. Hence, the fusion center can make global decisions about the PU state based on the outputs of the neural network. Relying on the final decision, if the channel is free, it is allocated to an SU (denoted as SU1) to transmit data. Furthermore, the SU is assumed to have a finite-capacity battery regularly recharged by a non-RF energy harvester. In addition to that, under energy constraint, the SU encrypts data using the AES algorithm with an appropriate key length to maximize the long-term security level of the system.

Regarding data protection techniques, there are two primary types of cryptography: symmetric (or *private key*) and asymmetric (or *public key*) algorithms. In general, using private-key cryptography for data encryption is not a time-consuming process, and thus expends less energy than public-key cryptography. For example, the experimental results from Kim *et al.* [30] showed that a public-key algorithm named the elliptic curve integrated encryption scheme (ECIES) consumes a thousand times more energy during the encryption process than the popular AES-128 private-key method. Even though a public-key algorithm can increase the security level by sacrificing a huge amount of energy, it is not a favorable choice for many wireless systems like CRNs. Subsequently, we focus on using the AES algorithm to secure the communications between SU1 and the FC. Specifically, the SU can use one of the three key sizes (128, 192, and 256) to encrypt data using the AES algorithm.

The security level is defined by the number of repetitions of the transformation rounds that convert the input data into encrypted data [31]. Therefore, the security level S_{Nk} is dependent on the key length Nk of the AES algorithm, as follows:

- $S_{Nk} = 10$ if $Nk = 128$ bits,
- $S_{Nk} = 12$ if $Nk = 192$ bits,
- $S_{Nk} = 14$ if $Nk = 256$ bits.

Using the longer key lengths provides the SU with better data security but consumes more energy [37]. As a result, at the beginning of each time slot, the SU decides its operation mode based on the sensing result and the remaining energy to maximize the long-term security level while efficiently using the limited energy. If the primary channel is busy, or the energy level in the SU's battery is too low, the SU stays silent to save energy for future use. If the channel is free and the remaining energy is acceptable, by calculating the total expected reward in future time slots, the SU can decide to keep silent to save energy, or to be active and transmit the data that is encrypted by the AES algorithm with a proper key length.

Regarding energy harvesting in the CRN, we assume that the SU operates based solely on wireless harvested energy that is stored in a finite-capacity battery. Hence, in designing network protocols, it is essential to obtain a reliable energy-harvesting model to guarantee energy autonomy in the network. The extensive experimental results from Lee *et al.* [35] showed that the transformed Poisson distribution model produces the nearest fit for most of the empirical datasets. The number of energy packets that an SU can harvest during a particular time slot, e_h, is given as

$$e_h \in \{e_{h,1}, e_{h,2}, \ldots, e_{h,max}\} \tag{2.3}$$

where $0 < e_{h,1} < e_{h,2} < \cdots < e_{h,max} < E_{ca}$, and E_{ca} is the battery capacity of the SU. We assume that e_h follows a Poisson point distribution with mean $e_{h,avg}$. Furthermore, the fit with the Poisson distribution can be improved by using a transformation $x = e_h - e_{h,min}$, where $e_{h,min}$ is the minimum harvested energy. The probability mass function (PMF) of e_h is then given by

$$P(e_h) = P(x = e_h - e_{h,min}) = \frac{e^{-x_{avg}} x_{avg}^{(e_h - e_{h,min})}}{(e_h - e_{h,min})!} \tag{2.4}$$

where $x_{avg} = e_{h,avg} - e_{h,min}$ is the sample average of the new variable x. This new distribution is called the transformed Poisson distribution (TPD). This transformation of the original variable can improve the fitting to the empirical datasets, as proven in [35]. In practice, although it is not easy to measure the exact amount of harvested energy in a

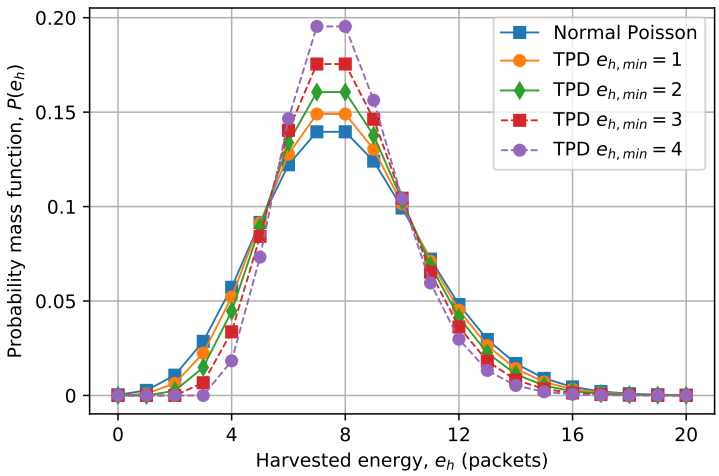

Figure 2.3: Comparison between the normal Poisson distribution and the transformed Poisson distribution with $e_{h,avg} = 8$ and different values of $e_{h,min}$.

time-slot interval, we can always estimate the average, the minimum and the maximum values of the harvested energy. Meanwhile, if the normal Poisson point process is used, the minimum harvested energy is assumed to be 0 (or zero) by default, which is rarely true in practical scenarios. For simulation purpose, the maximum value of harvested energy can be approximately determined if its cumulative distribution function is close enough to 1.

Figure 2.3 shows the difference in the PMF between the normal Poisson distribution and the transformed Poisson distribution when the average harvested energy is $e_{h,avg} = 8$ packets, with different values of minimum harvested energy: $e_{h,min} \in \{1, 2, 3, 4\}$ packets. As can be seen from the figure, the SU can harvest with a higher probability those energy values located near the mean by using the transformed Poisson model. As a consequence, we can also improve the learning rate of the actor–critic algorithm because the SU can focus on learning the variations of the energy values that are adjacent to the mean.

2.3 CNN-Based Cooperative Spectrum Sensing

In this section, we exploit the strength of the convolutional neural network, a particular type of deep neural network, to design a new cooperative spectrum sensing solution for the FC to determine the state of the PU on the primary channel. The process of

cooperative spectrum sensing is illustrated with the following steps:

1. The FC trains the CNN using historical sensing data represented by the local spectrum decisions provided by the SUs.

2. At the beginning of each time slot, all the SUs are required to perform local spectrum sensing by using an energy detection method and reporting their sensing outcomes to the FC via a control channel.

3. The FC uses the new sensing data as input for the trained CNN to make a global decision about the PU state on the channel of interest, and then feeds back the final decision to the SUs.

Accordingly, the problem of neural network-based cooperative spectrum sensing is divided into two important parts: local spectrum sensing by the SUs and global decision making by the FC using the trained CNN.

2.3.1 Local Spectrum Sensing

The considered CRN is assumed to be composed of K SUs. Each of them performs spectrum sensing independently using an energy detection algorithm, and then sends the outcome to the FC. Moreover, we assume that the status of the PU remains unchanged during each time slot. The hypothesis test statistics for local spectrum sensing at SU i can be formulated as follows [38]:

$$\begin{cases} A: & x_i(t) = h_i s(t) + w_i(t) \\ \bar{A}: & x_i(t) = w_i(t) \end{cases} \quad \forall i \in \{1, 2, \ldots, K\} \tag{2.5}$$

where $x_i(t)$ is the received signal by the ith SU in time slot t, h_i denotes the channel gain of the link between the PU and the ith SU, $s(t)$ denotes the PU signal, and $w_i(t)$ is zero mean and unit variance additive white Gaussian noise (AWGN).

Regarding energy detection, the observed energy at the ith SU is expressed as follows [39]:

$$xE_i = \sum_{j=1}^{N_i} |x_i(j)|^2; \ \forall i \in \{1, 2, \ldots, K\}, \tag{2.6}$$

where $x_i(j)$ is the jth sample of the received PU signal at the ith SU, and N_i is the number of sensing samples during each sensing period. For simplicity, we assume that the number

of sensing samples collected by each SU is the same for all the SUs. When N_i is sufficiently large (e.g., $N_i \geq 200$), xE_i can be approximated by a Gaussian random variable under the two hypotheses (A and $\bar{A}$) with mean μ_A, $\mu_{\bar{A}}$ and variance σ_A^2, $\sigma_{\bar{A}}^2$, given as follows [40]:

$$xE_i \sim \begin{cases} \mathcal{N}\left(\mu_A = N_i(1+\gamma_i), \sigma_A^2 = 2N_i(1+2\gamma_i)\right), & A \\ \mathcal{N}\left(\mu_{\bar{A}} = N_i, \sigma_{\bar{A}}^2 = 2N_i\right), & \bar{A} \end{cases} \tag{2.7}$$

where γ_i is the average gain of the sensed channel in terms of signal-to-noise ratio (SNR). We assume that γ_i follows a Gaussian distribution with mean μ_i and variance σ_i^2 as $\gamma_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$.

For a single-SU spectrum-sensing scheme, the local decision, D_i, is given by

$$D_i = \begin{cases} 1, & if\ xE_i \geq \lambda_i \\ 0, & otherwise \end{cases} \tag{2.8}$$

where 1 and 0 are single-bit data that represent states A and $\bar{A}$ of the primary user, respectively; and λ_i is a predefined decision threshold.

2.3.2 Global Decision Making

In a deep-learning research, the CNN is widely used in computer vision fields, such as image classification, speech recognition, and handwriting recognition, by making use of spatial characteristics. In this section, we present the process of creating and training a CNN for making a global decision about the PU state using the local sensing data as input.

Network Configuration

The first step in designing a CNN is to define the network layers that specify the structure of the CNN, as depicted in Figure 2.4. This network consists of the following layers [41].

- The *input* layer stores the input sensing data in the form of a gray scale image with size $1 \times K \times 1$, where K is the number of secondary users.
- The *convolutional (CONV2D)* layer contains K *neurons* that connect to the local subregions of the input image to learn its features by scanning through it. Each region has a size of 1×2.

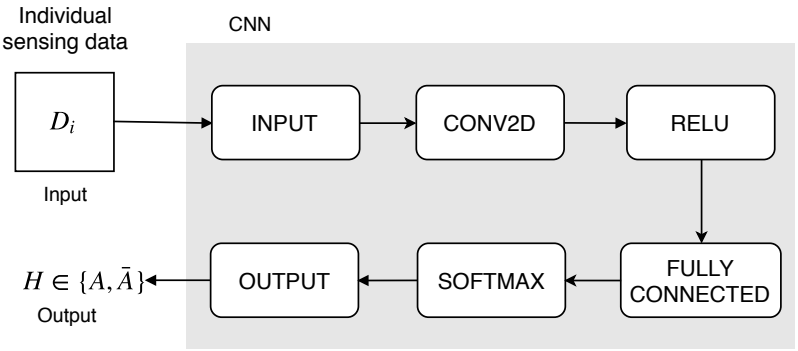

Figure 2.4: The structure of the CNN for cooperative spectrum sensing in CRNs.

- The *rectified linear unit (ReLU)* layer uses the ReLU function to introduce nonlinearity to the CNN by performing a threshold operation on each input element, simply defined as

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2.9}$$

- The *fully connected* layer combines all the local information from the original image (e.g., the results of feature extraction) determined in the previous layers to classify the status of the PU, which is active (A) or inactive ($\bar{A}$). Consequently, the size of the output data is equal to the number of states of the primary user.

- The *softmax* and *output* layers follow right after the fully connected layer for the classification problem. The softmax layer uses an output unit activation function, also known as a *normalized exponential* function, to create a categorical probability distribution for the two input elements (A and $\bar{A}$), as follows:

$$P(H_i) = \frac{\exp(q(H_i))}{\sum\limits_{H_j \in \{A, \overline{A}\}} \exp(q(H_j))}, \ i = 1, 2 \tag{2.10}$$

  where $P(H_i)$ is the class prior probability; $H_i \in \{A, \bar{A}\}$ is an element class; and $q(H_i)$ is the output value from previous layer of the sample given class H_i. Thereafter, the output (or *classification*) layer takes the values from the softmax function and assigns each input to one of the two classes.

It should be noted that the original image with size $1 \times K \times 1$ is a vector containing the local decisions from K SUs; thus, a one-dimensional (1D) convolution layer can be used

in the CNN to solve the problem of PU state classification instead of using a two-dimensional (2D) convolution layer. However, using a 2D CNN is more useful than 1D CNN in image classification. Furthermore, it would be easier to further develop the current approach to deal with three-dimensional data without making many changes in the current architecture of the CNN. For this reason, the size of the input image is generalized as $1 \times K \times 1$; thus, if the number of secondary users cooperating in spectrum sensing is large enough, the image size could be changed to $M \times N \times 1$, where $M \times N = 1 \times K$. Moreover, we can enhance the sensing accuracy by placing other information (e.g., the channel SNRs, the distances between the SUs and the PU) in the second and the third layers of the image, and performing some modifications (e.g., permutation, repetition) to the original data structure to provide the CNN with more features.

Network Training and PU Status Prediction

The local sensing decisions from the SUs, $D_i \ \forall i \in \{1, 2, \ldots, K\}$, are used as input for the CNN. Because a CNN is mostly used for image classification, the local decisions from K secondary users are rearranged to form a grayscale image with the size of $1 \times K \times 1$, where the last figure describes the number of color channels in the image. A stochastic gradient descent (SGD) optimizer with an adaptive learning rate is used in training the network. With this algorithm, the initial learning rate of 0.01 is later reduced based on a pre-defined schedule. For instance, it can be multiplied by a factor of 0.1 after every 10 epochs. The training set is a collection of local decisions from K SUs under different environmental conditions (i.e., a wide range in the sensed channel gain).

The FC uses the historical sensing data to train the CNN for the classification problem in advance. Thereafter, the FC determines the presence of the primary user on the licensed channel in every time slot by using the new individual sensing outcomes received at the beginning of each time slot as input for the trained network.

2.4 Energy-Efficient Data Encryption Schemes

In this section, we present two data protection schemes for an SU to enhance data security and energy utilization by determining its operation mode based on the current energy level in the battery and the sensing result. We assume that the SU always has enough energy for spectrum sensing, and that the SU determines its operation mode at the

beginning of each time slot. In particular, if the SU does not have enough energy to transmit data, or if the sensing result indicates the PU is in state A, the SU will stay silent during the remainder of the time slot. Otherwise, it can decide to transmit the data encrypted by the AES algorithm with one of the three key lengths, $Nk \in \{128, 192, 256\}$, considering the effect of the decision on the long-term security level of the system.

2.4.1 Markov Decision Process

The problem of the operation mode decision is first formulated as the framework of a Markov decision process that is defined as a tuple $\langle \mathbb{S}, \mathbb{A}, \mathbb{P}, \mathbb{R} \rangle$, where $\mathbb{S}$ is the state space, $\mathbb{A}$ is the action space, $\mathbb{P} : \mathbb{S} \times \mathbb{A} \mapsto \mathbb{S}$ is a transition probability function, and $\mathbb{R}$ is the reward function. The state of the SU at the tth time slot is defined as $s(t) = (e_r(t), \rho(t)) \in \mathbb{S}$, where $e_r(t)$ is the remaining energy of the SU, and $\rho(t)$ is the probability (also called *belief*) that the PU is inactive in that time slot. The action state space is defined as $\mathbb{A} = \{ID, TR_{Nk}\}$. At the tth time slot, the SU can choose to stay idle (action $a(t) = ID$) or it can choose to transmit data encrypted by the AES algorithm with key length $Nk \in \{128, 192, 256\}$ (action $a(t) = TR_{Nk}$). This action provides the SU with an immediate reward, $R(t)$, and causes the SU to transit into a new state, $s(t+1)$.

The reward (i.e., security level) achieved at the tth time slot when the SU is in state $s(t)$ and taking action $a(t)$ is defined as

$$R(t) \in \{0, S_{Nk}\} \tag{2.11}$$

where

- $R(t) = 0$ if the SU stays idle, or the transmission is not successful.
- $R(t) = 10$ if the transmission is successful, and the data are encrypted by AES-128.
- $R(t) = 12$ if the transmission is successful, and the data are encrypted by AES-192.
- $R(t) = 14$ if the transmission is successful, and the data are encrypted by AES-256.

The value function is defined as the total discounted reward from the current time slot ($t = 0$), when the SU's state is $s(t) = s$, which is given as follows [1]:

$$V(s) = \sum_{t=0}^{\infty} \eta^t R(t) | (s(0) = s) \tag{2.12}$$

where η is the discount factor. We aim to find the optimal action for the SU in the current time slot to maximize the value function as

$$a^*(0) = \arg\max_{a(t)\in\mathbb{A}} \left\{ \sum_{t=0}^{\infty} \eta^t R(t) | (s(0) = s) \right\} \tag{2.13}$$

The solution to the problem of the operation mode decision can be found by solving this equation.

2.4.2 A POMDP-Based Approach

In this part, we present the mode decision policy for the current time slot based on the POMDP framework. After taking action $a(t)$, the SU receives an instant reward, $R(t)$, and transforms to a new state, $s(t+1)$, which can be updated based on the following observations and transition probabilities.

Idle mode

If the SU decides to stay in idle mode (i.e., $a(t) = ID$), no reward is achieved, as $R(t) = 0$. In our work, we assume that the SU always has enough energy in its battery for spectrum sensing. Therefore, the remaining energy of the battery for the next time slot can be updated as follows

$$e_r(t+1) = \min(e_r(t) + e_h(t) - E_s, E_{ca}) \tag{2.14}$$

with the transition probability

$$P(e_r(t+1)|e_r(t), ID) = P(e_h(t)) \tag{2.15}$$

where $P(e_h(t))$ is the probability that the SU can harvest $e_h(t)$ energy packets in time slot t. The belief that the PU is inactive in the next time slot is given as

$$\rho(t+1) = \rho^*(t) P_{\bar{A}\bar{A}} + (1 - \rho^*(t)) P_{A\bar{A}}, \tag{2.16}$$

where $\rho^*(t)$ is the updated belief about the current time slot, which is calculated based on two observations at the end of the time slot, as follows.

a) Observation ϕ_1: The primary channel is sensed to be busy with probability

$$P(H(t) = A|\rho(t)) = \rho(t)P_f + (1 - \rho(t))P_d \tag{2.17}$$

where $H(t)$ denotes the state of the PU by spectrum sensing in time slot t. Then the belief in the current time slot is updated using Bayes' Rule as

$$\rho^*(t) = \frac{\rho(t)P_f}{\rho(t)P_f + (1 - \rho(t))P_d} \tag{2.18}$$

b) Observation ϕ_2: The sensing result indicates that the primary channel is free with the following probability

$$P(H(t) = \bar{A}|\rho(t)) = \rho(t)(1 - P_f) + (1 - \rho(t))(1 - P_d) \tag{2.19}$$

The belief about the current t^{th} time slot needs to be updated as follows:

$$\rho^*(t) = \frac{\rho(t)(1 - P_f)}{\rho(t)(1 - P_f) + (1 - \rho(t))(1 - P_d)} \tag{2.20}$$

Transmission mode

If the sensing result indicates that the PU is absent from the primary channel with the probability that is given in Equation (2.19), the SU can change to transmission mode when it has enough energy for data communications. In this mode, the SU transmits the data that is encrypted using the symmetric key algorithm with a proper key length, Nk, to maximize the effective security level. The remaining energy for the next time slot is updated as

$$e_r(t + 1) = \min(e_r(t) + e_h(t) - e_{tr}(t) - E_s - E_{Nk}, E_{ca}) \tag{2.21}$$

with the transition probability

$$P(e_r(t + 1)|e_r(t), TR_{Nk}) = P(e_h(t)) \tag{2.22}$$

The reward and the belief will be updated according to the acknowledgement (ACK) feedback signal that can be received from the SU recipient after the transmission is finished, as described in the following situations.

c) Observation ϕ_3: The SU transmitter receives an ACK message confirming that the transmission is successful, with the probability of correct detection as $p(t)(1 - P_f)$. Then, the reward for this case is

- $R(t) = 10$ if $Nk = 128$ bits
- $R(t) = 12$ if $Nk = 192$ bits
- $R(t) = 14$ if $Nk = 256$ bits

The belief that the channel will be free in the next time slot can be updated as

$$\rho(t+1) = P_{\bar{A}\bar{A}} \tag{2.23}$$

d) Observation ϕ_4: The transmission is unsuccessful (i.e., no ACK feedback is received), which means that a missed-detection event has occurred, with the probability $(1-\rho(t))(1-P_d)$; then, there is no reward: $R(t) = 0$. The belief that the channel will be vacant for the next time slot is given as

$$\rho(t+1) = P_{A\bar{A}} \tag{2.24}$$

Based on those observations, the optimal mode decision policy in Equation (2.13) can be rewritten as

$$a^*(0) = \arg\max_{a(t)} \left\{ \sum_{t=0}^{\infty} \eta^t \sum_{\phi_i} P(\phi_i) \sum_{e_r(t+1)} P(e_r(t+1)|e_r(t)) \times R(t)|(\phi_i, s(0)) \right\} \tag{2.25}$$

The final decision can be found to maximize the security level of the CR system by solving this equation using a *value iterations* method [42].

2.4.3 A Transfer Learning Actor-Critic Approach

In previous section, we propose a POMDP-based approach to solving the problem in Equation (2.13) on the assumption that the SU already has information about the harvested energy model. In this section, we introduce a new solution to the problem based on the actor–critic learning framework, which does not require the SU to already know the dynamics of energy harvesting. Instead, the SU determines those dynamics by directly interacting with the environment. A regular actor–critic model comprises three main elements: an actor (related to a learning policy), a critic (related to a learning value function), and the environment, as shown in Figure 2.5.

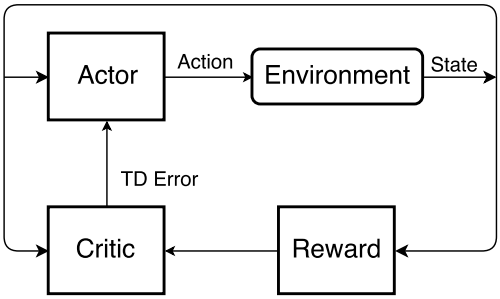

Figure 2.5: A regular actor–critic model. TD: temporal difference.

At time step t, the actor selects action $a(t)$ based on the current state, $s(t)$, and the policy, $\pi(s(t))$, which is defined by using a Gibbs softmax function as follows [1]:

$$\pi(s,a) = P(a(t) = a | s(t) = s) = \frac{e^{\theta(s,a)}}{\sum\limits_{a' \in \mathbb{A}} e^{\theta(s,a')}} \tag{2.26}$$

where $\theta(s,a)$ is the tendency to select action a when the SU is in state s. The final objective is to find an optimal mode decision policy for the SU at the tth time slot, and the problem in Equation (2.13) can be rewritten as

$$\pi^*(s) = \underset{a \in \mathbb{A}}{\operatorname{argmax}} \left\{ R(s,a) + \eta \sum_{s' \in \mathbb{S}} P(s'|s,a) V(s') \right\} \tag{2.27}$$

where $P(s'|s,a)$ is the transition probability from state s to state s' after taking action a.

After that, the SU transits into a new state, $s(t+1)$, and receives an instant reward $R(s(t), a(t))$. The critic evaluates the new state and computes a temporal difference (TD) error as

$$\delta(t) = R(s(t), a(t)) + \eta V(s(t+1)) - V(s(t)) \tag{2.28}$$

The critic uses the TD error to improve the estimate of the value function as well as the policy. The value function is updated as

$$V(s(t)) \leftarrow V(s(t)) + \alpha_c \cdot \delta(t) \tag{2.29}$$

where α_c is a positive parameter of the critic. The action resulting in a positive TD error is favorable, since the state value is better than expected. Hence, the probability of selecting action $a(t) = a$ in state $s(t) = s$ in the future should increase, and vice versa. Following that, the tendency to select this action is updated as

$$\theta(s(t), a(t)) \leftarrow \theta(s(t), a(t)) + \alpha_a \cdot \delta(t) \tag{2.30}$$

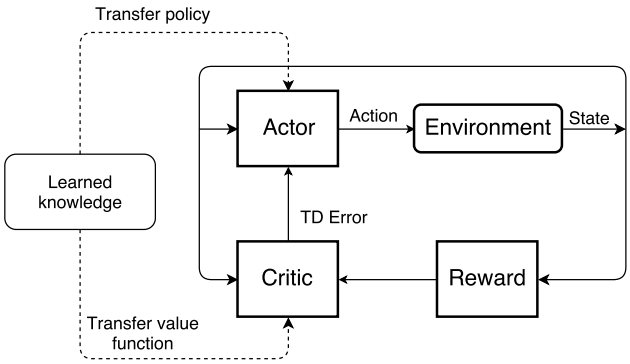

Figure 2.6: The transfer learning actor–critic model.

where α_a is a positive parameter of the actor. Furthermore, we exploit the idea of transfer learning to increase the convergence speed to the optimal solution by making use of historical learning data, as depicted in Figure 2.6.

The obtained information is transferred to the new actor–critic algorithm for real-time training in which the initialized value function is the same as the transferred function while the overall policy, $\theta_o(s(t), a(t))$, for choosing an action at time step t is given as

$$\theta_o(s(t), a(t)) = \varepsilon(t)\theta_l(s(t), a(t)) + (1 - \varepsilon(t))\theta_n(s(t), a(t)) \tag{2.31}$$

where $\theta_l(s(t), a(t))$ is the transferred policy; $\theta_n(s(t), a(t))$ is the new policy, which will be updated in every time slot by using Equation (2.30); and $\varepsilon(t)$ is the transfer rate, which will be reduced after each time step to gradually remove the effect of the transferred policy on the new one.

The training process of the actor–critic learning framework for the SU to decide its operation mode is illustrated as follows. At the beginning of the tth time slot, the SU chooses an action according to policy π considering the sensing result and the remaining energy in its battery. The SU can decide to stay idle, $a(t) = ID$, to save energy, or it can transmit the encrypted data, $a(t) = TR_{Nk}$, to the FC. The immediate reward, $R(s(t), a(t))$, and the next state, $s(t+1)$, are updated at the end of the time slot based on the observations that are presented in the POMDP-based scheme. Thereafter, the value function and the new policy are updated based on the received reward and the new state. This process repeats until it converges into the optimal solution that maximizes the long-term reward of the system, which means that value function $V(s)$ and policy $\pi(s)$ will finally converge to $V^*(s)$ and $\pi^*(s)$ as $k \to \infty$ [43].

2.5 Performance Evaluation

In this section, we present simulation results to demonstrate the efficiency of the proposed CNN-based cooperative spectrum sensing and data protection schemes in CRNs. We first present simulation results to evaluate the performance of the proposed CBCSS technique compared with other fusion techniques, such as a half-voting rule [44], an energy detection (ED) method performed by a secondary user, and the Chair–Varshney rule [45]. We then investigate the potential of the TLAC solution for establishing an operation mode decision policy by comparing it with the POMDP-based solution, the myopic scheme, and the fixed encryption methods, which will be described in detail later.

2.5.1 CNN-Based Cooperative Spectrum Sensing (CBCSS)

The proposed CBCSS for the two-state classification problem was implemented using the Neural Network Toolbox in Matlab (R2017a, The MathWorks Inc., USA, 2017). Unless presented otherwise, the simulation parameters were as listed in Table 2.1.

Table 2.1: Simulation parameters for the CBCSS scheme.

Symbol	Description	Value
K	The number of SUs	10
N_i	The number of sensing samples collected by each SU	300
γ_i	Average SNR of the sensed channel (dB)	-16 to -6
$P_{A\bar{A}}$, $P_{\bar{A}A}$	PU state transition probabilities	0.2

The average SNR of the sensed channel, γ_i, that was used for training the CNN ranged from $-16dB$ to $-6dB$. Furthermore, the number of training samples for each SNR was 2000. We consider three different performance metrics: probability of detection P_d, probability of false alarm P_f, and sensing error P_e. The total number of time slots for testing the performance of the proposed CBCSS was 10,000. Furthermore, the process was performed 10 times to get average values for P_d, P_f, and P_e. The first two parameters are calculated by using Equations (2.1) and (2.2), whereas sensing error is defined as the sum of the probability of false alarm (P_f) and the probability of missed detection ($1 - P_d$), as follows:

$$P_e = P_f + (1 - P_d). \tag{2.32}$$

In Figures 2.7 and 2.8, we compare the performance of the proposed CBCSS with

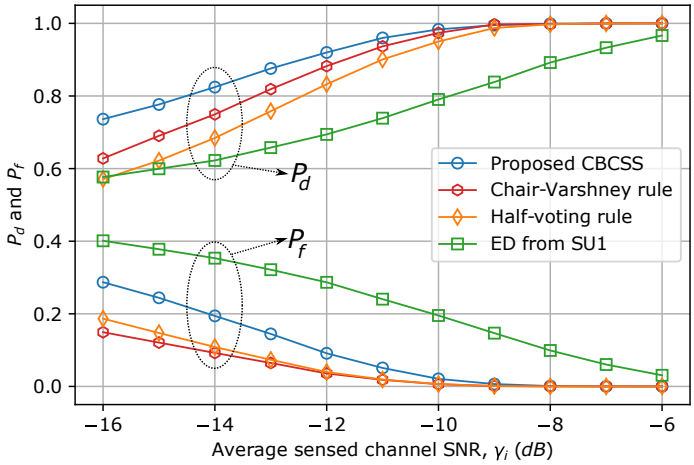

Figure 2.7: Probabilities of detection and false alarm according to average SNRs of sensed channel for different sensing schemes.

those of the conventional half-voting fusion rule for cooperative spectrum sensing, the local sensing result based on the energy detection method from one of the $K = 10$ secondary users, and the Chair–Varshney fusion rule. Regarding the half-voting rule, the fusion center makes a global decision based on the local sensing data. Specifically, the FC decides that the PU is active (A) if at least half of K SUs report the decision $D_i = 1$. With respect to the energy detection method, the local decision from SU1 was obtained for comparison. Under the Chair–Varshney rule, the detection statistics are expressed as the weighted sum of the local decisions; and the weights are functions of detection probability and false alarm [46]. The Chair–Varshney rule is the optimal decision fusion rule but requires a prior knowledge of the PU's activities and the local sensing performance of the secondary users.

From the figures, we can confirm that the proposed CBCSS outperforms other conventional methods, except for the Chair–Varshney optimal fusion rule, in terms of detection probability and sensing error. We can also see that with an increment in the average SNR, the probability of detection increases while the probability of false alarm and the sensing error decrease. This is because the effect of AWGN on the local decisions, and thus the training accuracy, decreases as SNR increases. Accordingly, larger sensed channel SNRs at the SUs provide better detection performance and fewer false alarms. Although the probability of false alarm with the proposed scheme is a little higher than with the half-voting and the Chair–Varshney rules, the total sensing error of the proposed CBCSS almost reaches to that of the Chair–Varshney optimal fusion rule and is lower than those of conventional

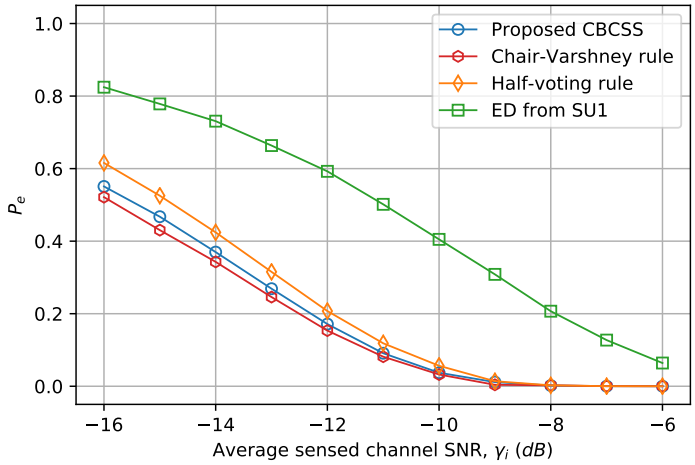

Figure 2.8: Sensing error according to average SNRs for different sensing schemes.

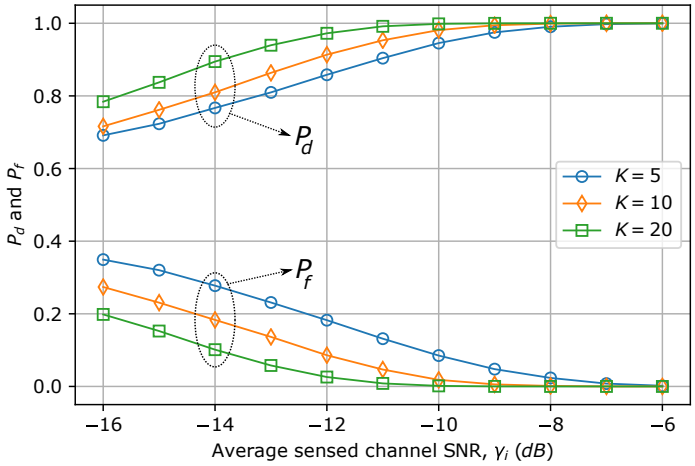

Figure 2.9: Probabilities of detection and false alarm with the proposed CBCSS according to average SNRs when the number of SUs, K, changes.

methods.

In Figures 2.9 and 2.10, we examine the effect of the number of secondary users, K, on the performance of the proposed CBCSS. To verify this, we evaluated the output results from three distinct CNNs that were trained with $K \in \{5, 10, 20\}$, while keeping the number of sensing samples unchanged at $N_i = 300$. For each value of K, the performance metrics were calculated again for comparison purposes. As can be seen from the figures, the increases in the number of SUs that cooperate in spectrum sensing can significantly improve the performance of the CBCSS. This is caused by the increase in spatial diversity

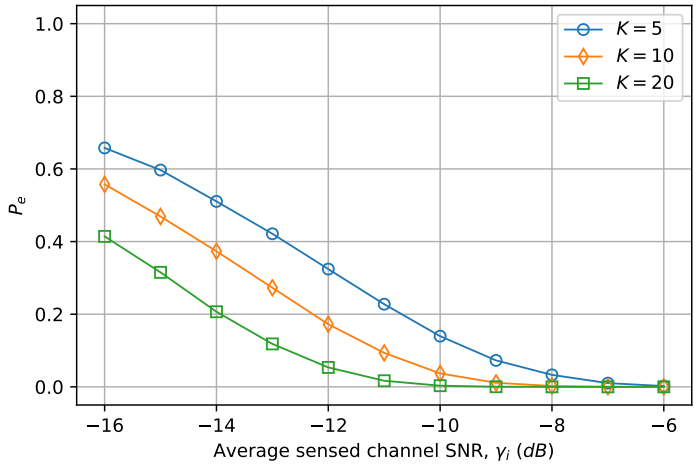

Figure 2.10: Sensing error with the proposed CBCSS according to average SNRs when the number of SUs, K, changes.

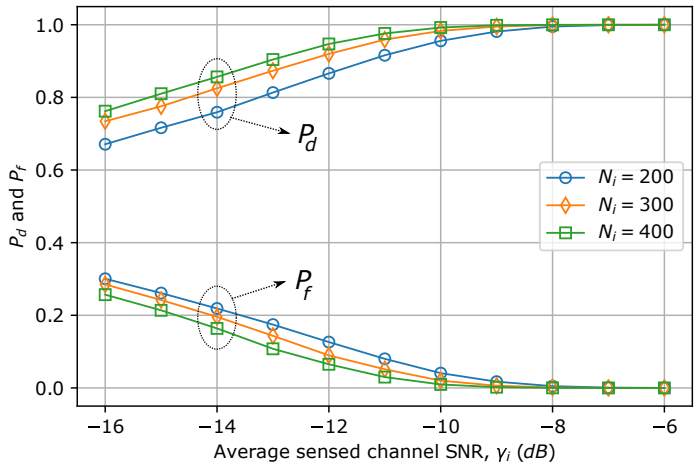

Figure 2.11: Probabilities of detection and false alarm according to average SNRs when the number of sensing samples for each SU changes.

when using more SUs, which can help the CNN to extract more information from the sensing data. Moreover, in Figure 2.10, there is almost no sensing error at $SNR = -10$ dB with $K = 20$ sensing nodes.

Finally, we measured the performance of the CBCSS by varying the number of sensing samples, N_i, as shown in Figures 2.11 and 2.12, for $K = 10$ secondary users. The training process is the same as with the changing K, but now the number of sensing samples is varied instead of K: $N_i \in \{200, 300, 400\}$. We assert that the effectiveness of the new cooperative spectrum sensing system can be improved by increasing the number of sensing

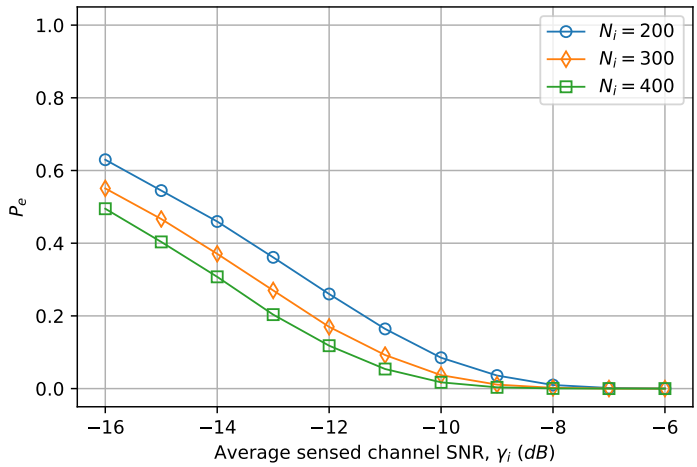

Figure 2.12: Sensing error according to average SNRs when the number of sensing samples for each SU, N, changes.

samples that are collected by the SUs for individual spectrum sensing using the energy detection method. Again, the larger value of γ_i provides better detection accuracy as well as a lower sensing error.

Since we focus on developing a new CNN-based cooperative spectrum sensing technique, for the sake of simplicity, we use a simple energy detection method for local spectrum sensing. However, the sensing efficiency can be further enhanced by improving the local spectrum sensing. That is, if the local sensing outcomes provide more accurate sensing data, the CNN can learn the features of the data with higher accuracy, which will produce more precise classification results. From the simulation results, we can observe that larger values of the channel SNR can ensure the better local sensing results, which leads to better overall sensing performance of the system.

2.5.2 Energy-Efficient Data Protection Schemes

This section verifies the performance of the proposed TLAC schem in comparison to the POMDP-based scheme, the myopic scheme, and AES algorithms with a fixed key length. With regard to the myopic scheme, if the PU is found absent from the channel, the SU will sacrifice its energy to maximize data security [47]. The POMDP framework requires complex numerical computations as well as prior information about the arrival of harvested energy. The complexity of the problem depends on the required amount of the computation space (e.g., the sizes of the input states, actions, transition probabilities, and observations).

Table 2.2: Simulation parameters

Symbol	Description	Value
γ_i	Average SNR of the sensed channel (dB)	-10
$P_{A\bar{A}}$, $P_{\bar{A}A}$	Transition probabilities between states (A and $\bar{A}$) of the primary user	0.2
E_{ca}	Battery capacity (packets)	160
E_s	Energy consumption for spectrum sensing (packets)	1
E_{Nk}	Energy consumption for data encryption using the AES algorithm with key length $Nk \in \{128, 192, 256\}$ (packets)	$\{4, 6, 8\}$
$e_{h,avg}$	Average harvested energy (packets)	$\{2, 4, 6, 8, 10\}$
e_{tr}	Energy consumption for data transmission (packets)	40
η	Discount factor	0.9
α_c	Critic learning rate	0.2
α_a	Actor learning rate	0.1
$\epsilon(0)$	Initial transfer rate	0.5
$\rho(0)$	Initial belief that the primary channel is free	0.5

In a POMDP, an agent controls the process by choosing the action at each time step based on the observation history to maximize the expected long-term reward. The optimal policy for the agent to choose an action can be found by solving the Bellman's equation using value iteration-based dynamic programming. Each iteration requires $O(|\mathbb{A}||\mathbb{S}|^2)$ operations to compute all the probabilities of transitioning from one state, $s \in \mathbb{S}$, to another state, $s' \in \mathbb{S}$, after taking an action, $a \in \mathbb{A}$. The actor–critic method, on the other hand, does not require the agent to compute all the occurrence probabilities to find the solution in advance. In addition to that, the agent learns the optimal policy from actual experienced transitions by directly interacting with the stochastic environment.

The basic simulation parameters for this exercise are shown in Table 2.2. For analytic convenience, we fixed the SNR value of the sensed channel at $-10dB$, and thus the probabilities of detection and false alarm are approximated as $P_d \approx 0.9$ and $P_f \approx 0.1$, respectively (based on the results of the proposed CBCSS method). We assume that the SU transmits a packet of 16-byte data in every time slot, which is equivalent to the minimum encryption block length in the AES cryptography; and the transmission channel gain is unchanged during a time slot. It is worth noting that one packet of energy is equivalent to $25\mu J$, and each simulation was run over a thousand of time slots for several iterations to obtain average values.

We first examined the convergence speed of the TLAC algorithm during the train-

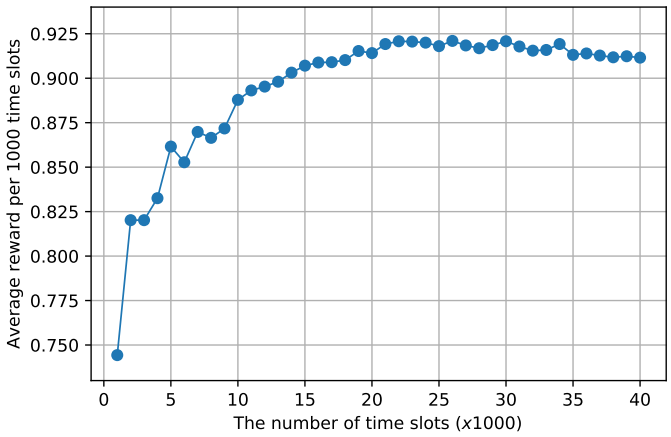

Figure 2.13: Actor–critic training convergence rate, $e_{h,avg} = 4$.

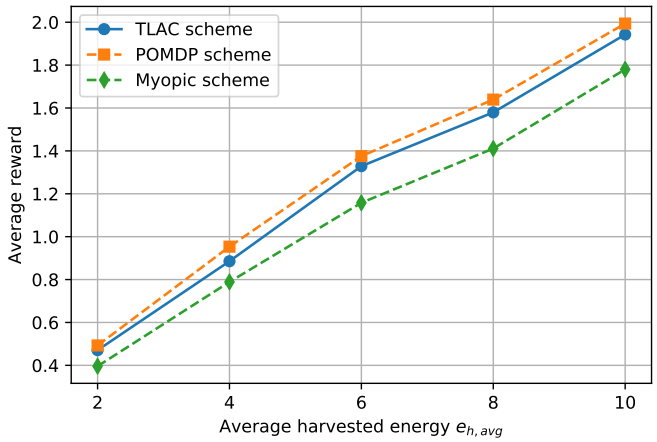

Figure 2.14: Average reward according to harvested energy for different data protection schemes.

ing process by calculating the average reward received after every 1000 time slots. The average harvested energy was fixed at $e_{h,avg} = 4$ packets. As can be seen from Figure 2.13, there is a significant rise in the convergence rate of the algorithm during the first 10,000 time slots of the training process; after that, the reward keeps increasing, but at a slower speed. Finally, the algorithm converges to an optimal policy for the SU to determine operation mode after 20,000 time slots when the reward is about 0.91.

In Figure 2.14, we show the efficiency of the proposed scheme compared with the POMDP-based and myopic schemes under the effect of harvested energy. As can be seen

from the figure, a larger harvested energy yields a higher reward, indicating that data are protected better. The reason is that, if the SU can harvest more energy, it has a greater chance to operate in transmission mode, and can transmit more data to the FC. Furthermore, the result of the proposed TLAC algorithm is better than the myopic one and a little lower than the POMDP method. To explain this, in the myopic scheme, the SU makes a decision on its working mode without considering the effect of this action on the future reward. In particular, if the primary channel is found free via spectrum sensing, the SU uses too much energy for data encryption to enhance data protection, which causes the SU to stay in idle mode over many time slots due to limited remaining energy. Regarding the POMDP-based solution, the SU is assumed to already have information about the harvested energy model, which is hardly ever true in practice. As a result, by using value iteration-based programming, we can compute all possible happening states and the corresponding occurrence probabilities to find the optimal policy beforehand. Consequently, the SU can predict the next state of the primary user and the upcoming harvested energy before effectively distributing the energy over future time slots. Meanwhile, employing the TLAC algorithm requires the SU to frequently interact with the environment to determine the dynamics of the arrival of harvested energy, which can result in a locally optimal policy [1]. In particular, the SU makes decisions based on a predefined policy (i.e., local or immediate consideration), which is updated at the end of every time slot, to improve future behavior without needing to have any information about the environment's dynamics.

Figure 2.15 illustrates the channel utilization by the SU for its data communications, computed as the ratio of the total number of successful data transmissions to the total time slots in which the primary user is sensed as inactive. From the figure, we can see that the primary channel is utilized more effectively when harvested energy $e_{h,avg}$ increases. In addition, the proposed TLAC algorithm utilizes the free channel better than the myopic scheme about 2% of the total successful transmissions. We can also see that the POMDP technique provides an optimal solution to the problem of the operation mode decision. However, the TLAC solution without requiring too much effort in mathematical computation or prior information about the environment's dynamics can provide the SU with a locally optimal policy that almost reaches the result of the POMDP scheme, especially when the amount of harvested energy is large. This is because the SU can encrypt data with a longer key size (e.g., $Nk = 256$) by utilizing extra energy in the battery when the average harvested energy increases. Therefore, the policy would be updated to favor

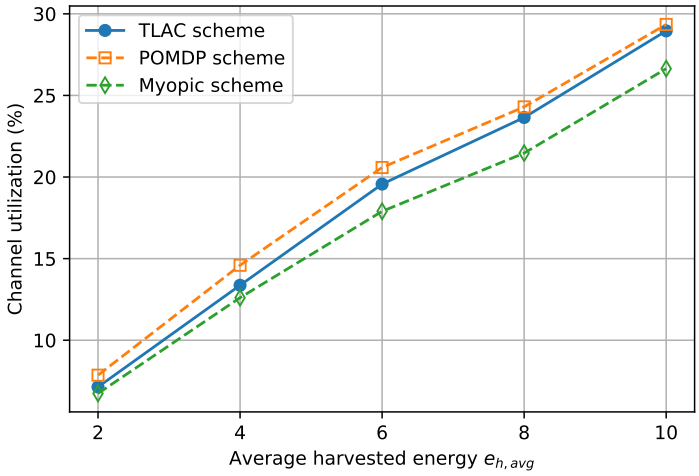

Figure 2.15: Channel utilization according to the harvested energy for different data protection schemes.

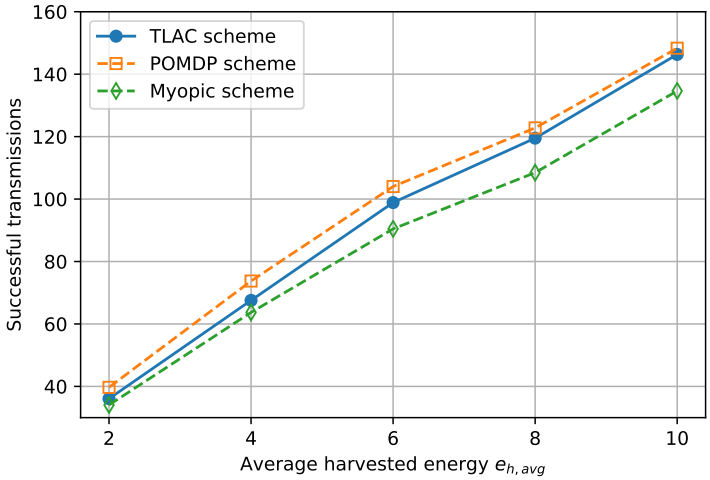

Figure 2.16: The number of successfully transmitted data packets according to the harvested energy for different data protection schemes.

the action that gives a better reward in the future.

Figure 2.16 depicts the total number of data packets transmitted from the SU to the fusion center based on harvested energy under three different data protection schemes. As can be seen from the figure, the SU can transmit more packets of data when using the TLAC algorithm, compared to the myopic scheme. The reason is that the proposed

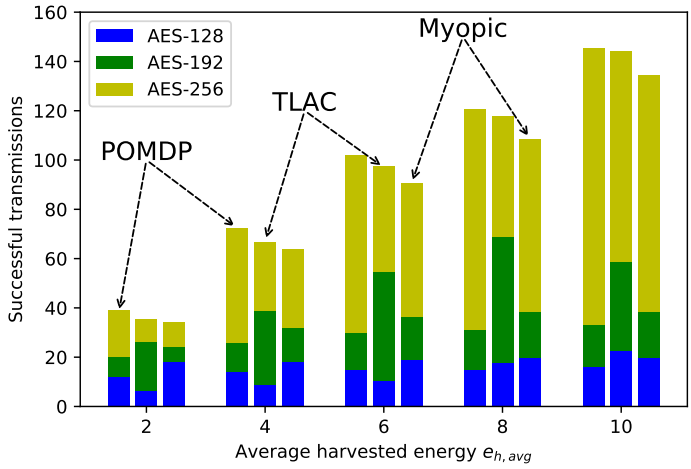

Figure 2.17: Comparison among the proposed schemes and the myopic scheme, based on harvested energy.

learning scheme can allocate the harvested energy more efficiently than the myopic one. Consequently, the SU can operate in transmission mode in more time slots, and thus, can transmit more encrypted data packets to the FC. Meanwhile, using the myopic scheme can cause the SU to be inactive due to lack of energy for future use. For that reason, the proposed TLAC framework can guarantee the security level, and can effectively utilize the limited energy resource. More specifically, in Figure 2.17, we present the detailed number of successfully transmitted data packets that are encrypted using the AES algorithm with different key lengths. We can see from the figure that the total number of data packets delivered under the TLAC algorithm is 10% higher than when using the myopic scheme, and that the POMDP scheme can provide the SU with the greatest number of transmitted packets. In particular, more packets are encrypted with longer key sizes (i.e., AES-192 or AES-256) with a rise in the arrival of harvested energy.

Finally, we examine the performance of the proposed schemes by comparing them with that of AES algorithms with fixed key length. In the fixed key length schemes, the SU uses only one key size to encrypt data at each time step even when it has enough energy. In Figure 2.18, the rewards under the proposed schemes and other schemes grow persistently with the increment in the harvested energy. While the proposed approaches provide the high average reward, the fixed encryption method with the shortest cipher key (AES-128) shows the lowest security level. The reason is that the proposed methods can efficiently

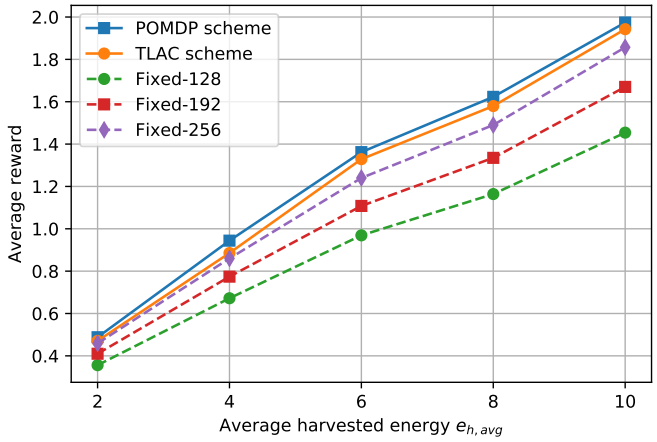

Figure 2.18: Reward comparison between the proposed schemes and the fixed key-length schemes according to the harvested energy.

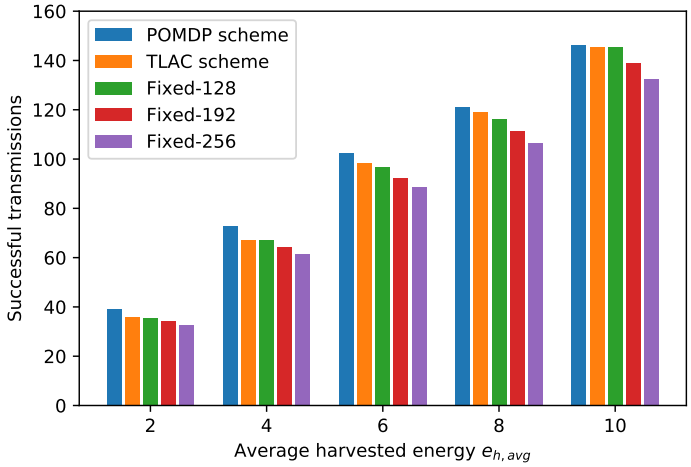

Figure 2.19: The number of successfully transmitted data packets according to the harvested energy, compared with the fixed key-length encryption methods.

allocate the energy to every time slot by estimating expected reward in the future time slots. Meanwhile, the AES-128 algorithm always uses the lowest amount of energy for data encryption, and thus, does not utilize the redundant energy in the SU's battery to enhance the security level as the arrival speed of the harvested energy increases.

On the other hand, the AES-256 uses maximum energy to encrypt data whenever the energy is sufficient to increase data security. However, this action reduces the chance

for the SU to operate in transmission mode, which leads to low successful transmissions, as shown in Figure 2.19. From the figure, we can see that the proposed POMDP scheme provides the SU with the highest channel utilization since the SU can transmit more data packets in comparison to other methods. This is because the fixed encryption techniques do not utilize the energy effectively for future use. Among those fixed encryption methods, the AES-128 with lower energy consumption allows the SU to transmit more data packets than the AES-192 and the AES-256, but provides the SU with the lowest reward. Consequently, we can verify that the proposed schemes can ensure effective data communications between the SU and the fusion center in terms of security level and channel utilization.

2.6 Conclusion

In this chapter, we present a CNN-based cooperative spectrum sensing and two energy-efficient data protection schemes in CRNs, by which the SUs can effectively utilize the primary channel under the constraint of limited harvested energy. We first design a new CNN-based cooperative spectrum sensing method. In this approach, the CNN is trained by using historical sensing data collected from secondary users under various environmental conditions. At the beginning of each time slot, the SUs individually perform spectrum sensing using an energy detection method, and then send the local decisions to a fusion center to make a global decision about the state of the primary user. The proposed CBCSS can increase the detection probability and remarkably reduce the sensing error, which can also contribute to effective communications between the SUs and the fusion center. Regarding the proposed data protection schemes, the SU determines its operation mode based on the remaining energy and the sensing result considering the effect of this decision on future time slots. By calculating the expected accumulated reward from the current time slot, the SU can decide to stay in idle mode to save energy for future use, or operate in transmission mode and transmit cypher data that are protected by using the AES algorithm with an appropriate key length. We then present simulation results to evaluate the performance of the proposed solutions, which show that the proposed schemes can guarantee energy-efficient data communications in cognitive radio networks.

Chapter 3

Optimal Power Allocation for Energy-efficient Data Transmission Against Full-duplex Active Eavesdroppers in Wireless Sensor Networks

3.1 Introduction

Wireless sensor networks (WSNs) are increasingly being deployed to monitor many sensitive and critical activities, and have become a promising solution to a wide range of applications. Typically, a WSN may contain a large number of compact, low-cost, and low-power wireless sensor (WS) nodes, which are connected through wireless channels to observe some phenomenon of the environment [48]. Furthermore, WSNs are normally deployed in unattended target areas; thus, the energy efficiency of WS nodes is always a crucial concern in order to guarantee self-sustainability and the lifetime of the nodes with respect to the energy required for operation, thereby having a significant impact on the performance of the entire network [49]. One of the most effective ways to improve the network lifespan is to use a small rechargeable battery integrated with an energy harvester to ensure energy autonomy, and thus, enable long-term and maintenance-free operation of the WS nodes. Recently,

wireless energy harvesting has become a promising technology for improving a battery's limited capacity and lifespan as renewable energy resources become available in many forms, including solar energy [50], wind power [51], thermal energy [52], and electromagnetic energy [53]. Therefore, it is essential to employ a self-sustaining scheme for energy autonomy in WSNs. For example, Valera *et al.* [54] characterized various existing environmental energy harvesting schemes that employ adaptive learning frameworks to achieve energy neutrality and maximize network performance in WSNs. Besides, Akhtar and Rehmani [55] described different efficient battery recharging techniques that not only extend the lifetime of a node but can also provide extra energy for enhanced functionality of the node.

Among the different types of renewable energy, solar power is one of the most common and effective energy resources in outdoor applications, and it can be scavenged from sunlight by using photovoltaic materials (i.e., solar cells). However, the solar power that can be harvested is highly dependent on environmental conditions like cloud, dust on the cells' surface, and illumination. In addition to solar power, radio frequency (RF) energy harvesting has recently become a promising solution for wireless communications networks due to the wide availability of radio sources (e.g., radio broadcasting towers, base stations, WiFi networks, and even mobile phones), which are not limited by space or time. An RF energy harvester can collect and convert radio signals into usable direct current (DC) voltage [56]. Furthermore, a crucial advantage of RF energy harvesting in WSNs is that a transmission from one WS node can provide power to all nodes that receive or listen to the transmission [57]. For this reason, Lee *et al.* [18] proposed a method for a primary wireless network to coexist with a secondary transmitter that harvests RF energy from transmissions by nearby primary transmitters while opportunistically accessing the licensed spectrum. The harvested energy is stored in a rechargeable battery with a finite capacity, which is then used for subsequent transmissions. More importantly, it is possible for a sensor node to integrate RF energy-harvesting modules with other energy-harvesting solutions, such as solar cells, to utilize the ambient energy [58].

Along with the emergence of low-powered wireless sensor networks, there has been growing consideration of wireless communications security [59]. The wireless signal, which is transmitted through open, random access, and shared wireless media, is easily vulnerable to malicious attacks by illegitimate users, such as data interception by an eavesdropper or transmission disruption by a jammer [60]. In this respect, physical layer security techniques in wireless networks have been widely studied as promising solutions to secure wireless

data transmissions, especially against eavesdropping [61]. Recently, full-duplex (FD) radio has become an emerging research topic for future wireless networks [62]. In contrast to half-duplex transmission, FD technology allows a radio to simultaneously transmit and receive information over the same frequency band at the same time, and thus, has the potential to double spectrum efficiency [63]. Therefore, a lot of research has been conducted considering the capability of FD communications, not only for spectral efficiency but also for enhanced security [64, 65]. On the other hand, adversarial users may also deploy FD technology to induce wireless security issues. To be specific, when operating in FD mode, an active eavesdropper is capable of jamming while eavesdropping to degrade the achievable transmission rate at the intended receiver [66]. To combat these issues, many existing studies focus on maximizing the secrecy rate in the presence of a powerful eavesdropper. In particular, Al-nahari [67] and Wu *et al.* [68] considered the effect of a massive multiple-input and multiple-output transmission strategy on the network secrecy rate in the presence of a multi-antenna FD eavesdropper. However, most of these works aimed to enhance network security without considering the impact of the current decision on the future performance of the network due to energy limitations.

Another important technology that has been increasingly used in WSNs is cognitive radio, by which cognitive users (i.e. unlicensed users) can opportunistically access licensed spectrum bands, solving the problems of spectrum scarcity and under-utilization. Furthremore, sensor nodes equipped with a cognitive radio can benefit the networks by increasing communications reliability and energy efficiency [11]. Thus, many studies in the literature discuss the application of cognitive radio in wireless communications to overcome the limitations of conventional WSNs [69, 70].

Inspired by these works, we investigate an energy-efficient and secure data transmission scheme against full-duplex active eavesdropper in a cognitive-aided wireless sensor network with energy harvesting and full-duplex communication. In this network, the eavesdropper opportunistically launches jamming attacks to further assist its eavesdropping process. In addition, each sensor node is equipped with a finite-capacity battery that can be recharged by both solar and RF energy harvesters. The legitimate destination with an FD capability can simultaneously receive data from the source and send an interference signal to the eavesdropper. The arrival of harvested energy at the sensor node and the jamming activity of the eavesdropper are modeled as a Poisson point process and a Markov discrete-time process, respectively. Moreover, one of the most energy-consuming components of a

sensor node is the wireless radio. Thus, it is essential for the node to set the transmit power to an appropriate level to effectively utilize its limited energy for communications. Therefore, we propose an optimal power allocation scheme for energy-efficient data transmission against FD active eavesdroppers to maximize the long-term secrecy capacity of networks under energy constraints. With this scheme, the sensor nodes in the network need to carry out cooperative spectrum sensing to decide whether the eavesdropper is conducting a jamming attack or not; then, a sensor node assigned to data transmission can either stay idle or transmit data to the destination using the appropriate transmit power. In particular, the main contributions of this work can be summarized as follows.

- We investigate a new model for energy-efficient data transmissions in WSNs in the presence of FD active eavesdroppers. In this model, the source node is powered by solar and RF energy harvesters; the eavesdropper always listens to the legitimate transmissions and opportunistically executes jamming attacks towards the legitimate destination. Meanwhile, the destination can also employ its FD capability to degrade the wiretap rate at the eavesdropper.

- We first formulate the problem of transmission security against FD eavesdropping as the framework of a partially observable Markov decision process (POMDP), and then use value iteration–based dynamic programming to find the optimal transmit power decision policy for a sensor node in order to maximize its long-term secrecy rate within the constraints of harvested energy.

- We further present another approach to solve the security problem by using a model-free reinforcement learning framework, namely, an actor-critic algorithm. With this approach, the sensor node directly interacts with the environment, and then, learns the optimal transmit power decision policy from trial-and-error experience.

The remainder of this chapter is organized as follows. In Section 3.2, we provide a literature review of related works. In Section 3.3, we present the system model and the original security problem. In Section 3.4 and Section 3.5, we describe the optimal and sub-optimal solutions to the security problem. In Section 3.6, we present the simulation results to evaluate the performance of the proposed schemes. Finally, the conclusion is provided in Section 3.7.

3.2 Related Works

Conventionally, research work on physical layer security is usually with the assumption of half-duplex transmission in which the security threats come from either eavesdropping or jamming at a time. One of the common approaches to combat eavesdropping (or jamming) is to deploy multiple antennas in signal beamforming [71] (or in jamming-aware decision fusion [23]). Besides multi-antenna techniques, other effective methods based on artificial-noise injection are also widely applied. For example, Lin *et al.* [72] proposed a generalized artificial noise scheme that allows injection of artificial noise into a legitimate channel to improve the secrecy rate. Additionally, it is also possible to guarantee wireless security against eavesdropping by relying on external support from friendly nodes. An example of this cooperative security is collaborative beamforming using multiple relays to achieve confidentiality [73]. Zou *et al.* [74] also considered physical layer security in cooperative wireless networks and examined the best relay selection scheme to improve wireless security against eavesdropping attacks.

On the other hand, many researchers have studied physical layer security regarding FD transmission capability. In particular, Zheng *et al.* [75] studied the potential benefits of an FD destination node simultaneously acting as a receiver and a jammer to enhance the secrecy rate. The work in [76] investigated a cooperative mechanism in which the relays work in FD mode to receive and forward data together with extra jamming signals. Mukherjee and Swindlehurst examined potential countermeasures against an active eavesdropper that intends to cause maximum disruption to the security of the main channel [77]. Tang *et al.* [78] employed a hierarchical game framework to formulate the security problem when facing a full-duplex active eavesdropper. More specifically, they investigated the optimal strategies for both legitimate transmission and wiretap rate maximization.

Due to the stochastic property of wireless channels and energy harvesting, the Markov decision process (MDP) and its variants have been used widely in solving stochastic optimization problems in low-power wireless networks [79, 80]. However, most of existing work assumes that the agent can obtain an accurate information about the environment's evolution, which is hardly true in practice. Therefore, the learning-based framework is a better choice and has been widely employed in wireless networks recently, in which the agent learns its optimal decision policy with no prior information about the environment's dynamics [81]. For example, Wei *et al.* [82] proposed an actor-critic learning algorithm for

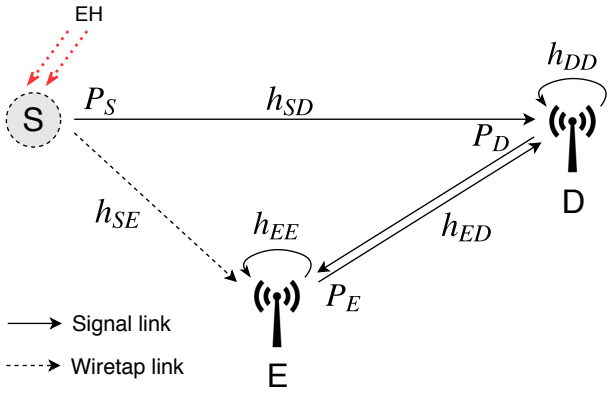

Figure 3.1: The considered system model. EH: energy harvesting.

efficient user scheduling and resource allocation in heterogeneous wireless networks powered by hybrid energy.

Different from the existing work on physical layer security against FD eavesdropper in wireless networks with energy harvesting, we propose two energy-efficient power allocation schemes, with the purpose of maximizing the long-term network performance for both deterministic and non-deterministic environment.

3.3 System Model

Wireless sensor networks are usually divided into clusters to increase energy efficiency and improve the scalability of the network [83]. Each cluster has a coordinator (or cluster head) that is responsible for gathering the aggregated data from other sensor nodes and sending it to the sink (or base station). An ideal cluster head is a node that has higher energy and better capability than the other sensor nodes. We consider a cluster in a WSN that consists of a cluster head, a hidden eavesdropper, and a number of sensor nodes. In our scenario, the cluster head is powered by conventional power grid energy, while the regular sensor nodes are powered by wireless energy sources, including solar power and RF energy. Herein, we investigate energy-efficient and secure communications between one of the regular sensor nodes and the cluster head in the presence of an active eavesdropper. We denote a regular sensor node as S, the cluster head as D, and the eavesdropper as E, as shown in Figure 3.1.

In this system, active eavesdropper E always tries to listen to the legitimate com-

munications between sensor node S and cluster head D on a wiretap link that is assumed to have channel gain h_{SE}. In addition, E can also conduct jamming attacks opportunistically to degrade the signal received at D. Regarding the jamming signal, we assume that the jamming power is P_E, and the jamming link gain is h_{ED}. However, the jamming process also affects the eavesdropper due to self-interference, which is difficult to suppress entirely. In this case, the self-interference link gain is denoted as h_{EE}, and we use a linear coefficient, η, as a self-interference attenuation factor.

For legitimate transmissions, we denote the gain of the signal link from sensor node S to cluster head D as h_{SD}, and the transmit power is P_S. Moreover, the cluster head is also assumed to have an FD capability to transmit and receive signal at the same time. Therefore, it can send artificial noise to the eavesdropper to deteriorate its signal-to-interference-plus-noise ratio (SINR) while receiving the desired data from the sensor nodes. We denote the self-interference link gain and the jamming power at the cluster head as h_{DD} and P_D, respectively. We also use the same linear coefficient η for the residual self-interference at D. The channel coefficient takes into account both distance dependence and shadow fading path loss. For simplicity, we will call the regular sensor node the source and the cluster head the destination, when no ambiguity arises. We also assume that the self-interference can be significantly suppressed, which means that the self-interference coefficient is sufficiently small. Hence, the impact of the residual self-interference on the received signal at the destination is controllable.

3.3.1 Cognitive-aided wireless sensor network

The eavesdropper is assumed to have a limited energy capacity, and thus, cannot always execute jamming attacks while eavesdropping. Therefore, we use a two-state Markov discrete-time process to model the full-duplex activity (i.e. a jamming attack) of the eavesdropper, as depicted in Figure 3.2. We assume that the eavesdropper operates in a time-slotted fashion, in which a time slot duration is denoted as T, and switches between jamming and not jamming according to static probabilities. The state transition probabilities are denoted as $P_{ij} : i, j \in \{J, \bar{J}\}$, where J and $\bar{J}$ represent jamming and not jamming, respectively. Moreover, from the security point of view, the legitimate nodes need to adapt to the jamming attack's dynamics by exploiting a cognitive capability and automatically modifying their parameters.

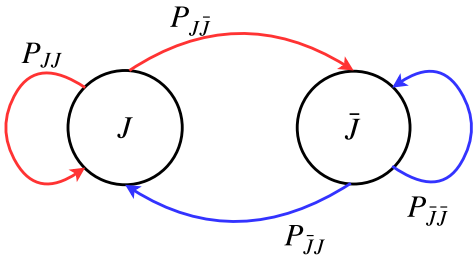

Figure 3.2: Eavesdropper's jamming attack model.

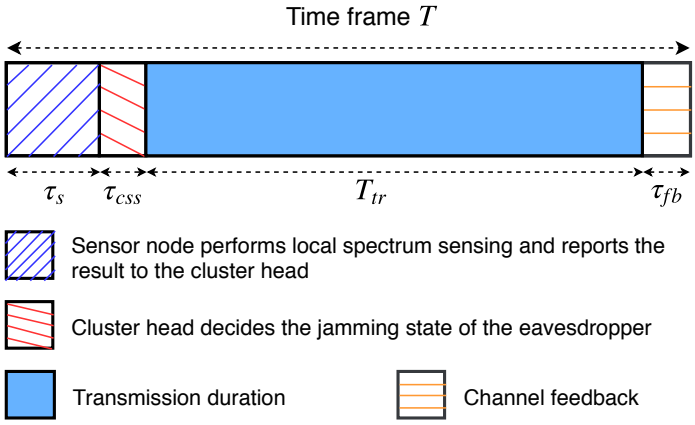

Figure 3.3: A time-frame structure for the cognitive operation of the system.

The cognitive operation proceeds as shown in Figure 3.3. At the beginning of each time slot, the cluster head directs all sensor nodes in the cluster to perform local spectrum sensing separately to detect a jamming signal on a selected channel. In response, each sensor node senses the presence of a jamming signal on the considered channel by using an energy detection technique [84], and then sends the outcome to the cluster head. This process consumes τ_s seconds, $0 < \tau_s < T$. We assume that the node always has enough energy for the sensing process, which includes local spectrum sensing and reporting the outcome to the cluster head, and that S always has data that it needs to send to D. The energy consumption for the sensing process in each time slot is assumed to be fixed at E_s packets of energy. At cluster head D, which is equipped with a fusion center, the local sensing data are combined using a certain fusion rule (e.g., soft combination scheme [85]) to decide the jamming state of the eavesdropper. The final decision on jamming activity is then broadcast to all nodes in the cluster; this process costs τ_{css} seconds. The performance

of the cooperative sensing scheme can be evaluated by the probability of detection (P_d) and the probability of false alarm (P_f). The former metric refers to the probability that the presence of a jamming signal is detected correctly, which is given by

$$P_d = \Pr\left(H_J^{(t)} = J | J\right) \tag{3.1}$$

where $H_J^{(t)}$ denotes the state of the jamming signal as informed by spectrum sensing at time instant t. The latter metric is the probability that the sensing result indicates the presence of a jamming signal when it is actually not there, which is given by

$$P_f = \Pr\left(H_J^{(t)} = J | \bar{J}\right) \tag{3.2}$$

In addition, every sensing method is developed with the purpose of keeping as low a number of false alarms as possible while guaranteeing a high detection rate [86]. Moreover, we assume that these metrics are available at the cluster head.

Based on the sensing result, if there is no jamming attack, the channel is assigned to a sensor node for data transmission. Therefore, we are going to investigate an optimal transmit power decision policy for secure data transmissions between one sensor node (i.e. the source) and the cluster head (i.e. the destination) against a hidden active eavesdropper. To be more specific, based on the sensing results and prior information about the arrival of harvested energy, as well as about the jamming attack model, the source can either stay idle to save energy or transmit data to the destination using an appropriate transmit power during the transmission time of the time slot, T_{tr}. In the meantime, the destination will also try to jam the eavesdropper with artificial noise.

3.3.2 Energy arrival models

Solar energy harvesting

The legitimate source node, S, is assumed to have a limited-capacity battery, E_{bat}, which is constantly recharged by a solar energy harvester; thus, the source can harvest solar energy and perform other operations simultaneously. Although the amount of harvested energy may vary for many reasons, we can always estimate the average number of energy packets that the nodes can harvest in a time slot. We consider a practical case, where the arriving harvested energy packets, e_h, during a short time interval follow a Poisson distribution with mean $E_{h,avg}$, as investigated in [35]. In particular, the number of energy

packets that the source can harvest during the t^{th} time slot is given as

$$e_h^{(t)} \in \{e_1, e_2, \cdots, e_\xi\} \tag{3.3}$$

where $0 < e_i < E_{bat},\ i \in \{1, 2, \cdots, \xi\}$; and the cumulative distribution function (CDF) is given as

$$F\left(e_h^{(t)}; E_{h,avg}\right) = \sum_{n=0}^{e_h^{(t)}} e^{-E_{h,avg}} \frac{(E_{h,avg})^n}{n!} \tag{3.4}$$

We assume that the source node always has enough energy to carry out basic operations (i.e., sending and receiving control signals, or activating the energy harvesting devices). Therefore, we further define an energy threshold, E_{th}, which could be used to determine the operation mode of the sensor node. To be specific, if the current energy level in the battery of the node is lower than the threshold, it stays idle and waits for more harvested energy. Otherwise, it can send data to the destination using a suitable number of energy packets. For simulation purposes, the maximum value of harvested energy can be approximately determined if its CDF is close enough to 1.

RF energy harvesting

Source node S is also equipped with an RF energy harvesting module (as a single system on chip) that consumes little power but supplies extra energy to the system by extracting DC power from received electromagnetic waves [87]. To reduce software complexity, S is assumed to have a single radio for a WSN scenario, which means that S can only use one radio for both RF energy harvesting and communicating. One major limitation of harvesting energy from RF resources is that RF power rapidly decreases over distance, which results in low available power for harvesting. Many measurements have shown that the power densities of a typical RF energy harvester range from $0.1\mu W/cm^2$ to $1mW/cm^2$ [55,56]. We assume that the source can harvest RF energy mostly from adjacent sensor nodes and the eavesdropper, as illustrated in [18]. Furthermore, the global information of the cluster (e.g. channel gain, node positions, and transmit power) are available at the cluster head, and thus, can be available to any node in the cluster. By carrying out RF energy harvesting, source S can obtain additional energy and increase the probability of detecting the jamming signal produced by the eavesdropper, particularly when S harvests RF energy during the whole slot duration (after carrying out spectrum sensing).

The RF-harvested energy packets in a time slot, denoted as $e_{rf}^{(t)}$, are assumed to be independent across time slots. We define an energy harvesting threshold, ϵ, to determine the number of RF energy packets that can be harvested in a particular time slot. At the t^{th} time slot, if the received power is too small to activate the energy harvesting circuit, the harvested energy is assumed to be negligible. Hence, the probability of harvesting $e_{rf}^{(t)}$ energy packets from RF sources is defined as

$$\Pr(e_{rf}^{(t)}) = \begin{cases} 1, & e_{rf}^{(t)} \geq \epsilon \\ 0, & otherwise \end{cases} \tag{3.5}$$

Regarding the jamming activity of E, if the transmit power of the eavesdropper is P_E, the total number of RF energy packets that can be harvested from the jamming signal during duration T_{tr} is given as follows [18]:

$$e_{rf}^{(t)} = \rho T_{tr} P_E |h_{SE}|^2 \tag{3.6}$$

where ρ is harvesting efficiency. It is worth noting that the jamming power of the eavesdropper is assumed to be high enough for deteriorating the legitimate signal at the destination. Furthermore, the distance between the eavesdropper and the sensor as well as among the sensors in the cluster are not too far, and thus, the signal attenuation is acceptable and controllable. Typically, RF energy harvesting efficiency is between 50% and 75% over a 100-meter range of input power [57]. And we assume that the RF energy harvested from the jamming signal of the eavesdropper is more dominant than that of the sensor nodes. In this work, solar energy is the main power supply of the source node, and for this reason, it is important to note that we use the word *harvested energy* instead of solar-harvested energy throughout this chapter for simplicity.

3.3.3 Full-duplex secrecy capacity

We assume that the destination can only decode the legitimate signal from the source when there is no jamming attack from the eavesdropper. Besides, the global channel state information is available to all nodes in the system at the beginning of each coherence interval (i.e. a time slot). We also assume that the source and destination are fully co-operative, and any data transmission from eavesdropper E to source S and destination D could allow them to estimate the required channel amplitude responses from S and D to

E via reciprocal characteristics. Furthermore, as the destination is provided with electrical energy by the traditional power grid, it always has enough energy to transmit and receive data. As a result, the SINR of the received signals at destination D and eavesdropper E are given as

$$\gamma_D = \frac{P_S|h_{SD}|^2}{\eta P_D|h_{DD}|^2 + \sigma_w^2} \tag{3.7}$$

and

$$\gamma_E = \frac{P_S|h_{SE}|^2}{P_D|h_{DE}|^2 + \sigma_w^2} \tag{3.8}$$

where η is the coefficient of self-interference, and σ_w^2 is white Gaussian noise power, which is assumed to be the same at D and E. Taking into account the path-loss component, we have $|h_{XY}|^2 = g_{XY} d_{XY}^{-\zeta},\ X \in \{S, D\},\ Y \in \{D, E\}$, where ζ is the path-loss exponent, d_{XY} is the distance between the nodes, and g_{XY} is an exponentially distributed random variable [64]. By letting $\Gamma_{XY} = \frac{P_X|h_{XY}|^2}{\sigma_w^2}$, the SINR at D and E, respectively, can be rewritten as

$$\gamma_D = \frac{\Gamma_{SD}}{\eta\Gamma_{DD} + 1} \tag{3.9}$$

and

$$\gamma_E = \frac{\Gamma_{SE}}{\Gamma_{DE} + 1} \tag{3.10}$$

By applying the Shannon capacity formula [88], we can obtain the transmission rates (*bits/sec/Hz*) at destination D and eavesdropper E, respectively, as

$$R_D = log_2(1 + \gamma_D) \tag{3.11}$$

and

$$R_E = log_2(1 + \gamma_E) \tag{3.12}$$

The secrecy rate can be defined as the difference between the rates of D and E:

$$\begin{aligned} R_S &= [R_D - R_E]^+ \\ &= \max\left(\log_2\left(\frac{1+\gamma_D}{1+\gamma_E}\right), 0\right) \end{aligned} \tag{3.13}$$

Above all, the primary purpose of this work is to find an optimal transmit power decision policy for source S to maximize the secrecy rate, as follows:

$$\begin{aligned} &\max_{P_S} R_S \\ &subject\ to\ 0 \le P_S \le P_S^{max} \end{aligned} \tag{3.14}$$

where P_S^{max} is the upper bound of the transmit power that the source can use to transmit data to the destination without causing it to be inactive due to lack of energy.

3.4 Optimal Power Allocation Scheme for Energy-Efficient Data Transmission Against FD Eavesdropper

In this section, we introduce an optimal transmit power decision policy for the source in order to maximize the secrecy rate based on prior information about the arrival of harvested energy and about the jamming activity. In particular, at the beginning of each time slot, after performing spectrum sensing, the source will send the local sensing result to the destination as well as information about its current energy level. The destination (i.e. the cluster head), with its better computation capability, will make the decision on transmit power based on the information, and feeds the decision back to the source via the control channel. To be more specific, if the source does not have enough energy for data transmission, or the sensing result indicates the existence of a jamming attack on the channel of interest, the source operates in RF mode to harvest energy from RF signals and waits for more energy from the solar harvester. Otherwise, it changes to data transmission mode and uses a suitable transmit power to maximize the expected secrecy rate in the long run. The transmit power decision policy problem is first formulated as the framework of a partially observable Markov decision process [89] in which the effect of the current time slot's decision on future time slots is considered, and it is then solved by using value iteration–based dynamic programming.

3.4.1 Markov decision process

The Markov decision process is usually defined as a tuple $\langle \mathbb{S}, \mathbb{A}, \mathbb{P}, \varphi \rangle$, where $\mathbb{S}$ is the state space, and $\mathbb{A}$ is the action space; $\mathbb{P} : \mathbb{S} \times \mathbb{A} \mapsto \mathbb{S}$ is the state transition function, and $\varphi : \mathbb{S} \times \mathbb{A} \mapsto \mathbb{R}$ is the reward function. We define the system state in the t^{th} time slot as $s^{(t)} = \left(e_{r,S}^{(t)}, \mu_E^{(t)}\right) \in \mathbb{S}$, where $e_{r,S}^{(t)}$ is the remaining energy of the source, and $\mu_E^{(t)}$ is the probability (also called the *belief*) that E is not conducting a jamming attack against D in that time slot, $\mu_E^{(t)} = \Pr(\bar{J})$. Furthermore, $e_{r,S}^{(t)}$ and $\mu_E^{(t)}$ are updated at the end of the time slot based on the selected action and the observations. We define the set of actions as $\mathbb{A} = \{0, e_{tr1}, e_{tr2}, \cdots, e_{tr\psi}\}$, where $0 < e_{tr1} < \cdots < e_{tr\psi} < E_{bat}$. At time instant t, source S uses finite packets of energy, $e_{tr,S}^{(t)} \in \mathbb{A}$, to transmit data to the destination based on the sensing result and the current system state. It is worth noting that the action $e_{tr,S}^{(t)} = 0$ indicates that the source is operating in RF mode.

We denote as $R(s^{(t)}, e_{tr,S}^{(t)})$ the reward achieved at the end of the t^{th} time slot when the source is in state $s^{(t)}$ taking action $e_{tr,S}^{(t)}$, which can be calculated as

$$R\left(s^{(t)}, e_{tr,S}^{(t)}\right) = \max\left(\log_2\left(\frac{1+\gamma_D^{(t)}}{1+\gamma_E^{(t)}}\right), 0\right) \tag{3.15}$$

where $\gamma_D^{(t)}$ and $\gamma_E^{(t)}$ are the temporal SINR at the destination and the eavesdropper, respectively, in the t^{th} time slot, respectively. SINR can be calculated by using Equation (3.9) and Equation (3.10) with a little modification in transmit power. More specifically, the transmit power of a node is computed using transmit energy $e_{tr,X}^{(t)}$ and transmission duration T_{tr}, as follows:

$$P_X^{(t)} = \frac{e_{tr,X}^{(t)}}{T_{tr}} \tag{3.16}$$

where $X \in \{S, D\}$.

The main objective of this work is to find the optimal transmit power policy for the source in the t^{th} time slot to maximize the accumulated reward from this time slot, which is described as follows:

$$e_{tr,S}^{*(t)} = \underset{e_{tr,S}^{(t)} \in \mathbb{A}}{\arg\max} \left\{ \sum_{k=t}^{\infty} \beta^{k-t} R\left(s^{(k)}, e_{tr,S}^{(k)}\right) | s^{(t)} \right\} \tag{3.17}$$

where $0 \leq \beta \leq 1$ is the discount factor, which signifies the effect of the future rewards on the current time slot.

3.4.2 Value iteration–based problem solution

In this section, we present the decision policy for the t^{th} time slot with the observations and transition probabilities to draw the immediate reward that is received at the end of the time slot, $R(s^{(t)}, e_{tr,S}^{(t)})$, and to update the system state for the next time slot, $s^{(t+1)}$, as follows.

Case 1

The sensing result indicates the presence of a jamming signal on the channel with probability

$$\Pr\left(H_J^{(t)} = J | \mu_E^{(t)}\right) = \mu_E^{(t)} P_f + (1 - \mu_E^{(t)}) P_d \tag{3.18}$$

The source trusts this result, and does not transmit data to the destination. Instead, it performs RF energy harvesting during the remainder of the time slot (i.e. $e_{tr,S}^{(t)} = 0$). In this case, there is no reward: $R(s^{(t)}, e_{tr,S}^{(t)} = 0) = 0$. The remaining energy of the battery that can be used for the next time slot is updated as follows:

$$e_{r,S}^{(t+1)} = \min\left(e_{r,S}^{(t)} + e_{h,S}^{(t)} + e_{rf}^{(t)}, E_{bat}\right) \tag{3.19}$$

with the transition probability given as

$$\Pr\left(e_{r,S}^{(t+1)} | e_{r,S}^{(t)}, e_{tr,S}^{(t)} = 0\right) = \Pr\left(e_{h,S}^{(t)}\right) \tag{3.20}$$

where $\Pr(e_{h,S}^{(t)})$ is the probability that the source can harvest $e_{h,S}^{(t)}$ packets of solar energy in the t^{th} time slot. The system state is updated based on the following observations.

Observation ϕ_1 The RF energy harvested in the time slot is less than a pre-defined threshold, $e_{rf}^{(t)} < \epsilon$, which means there is a false alarm about the jamming attack, with probability

$$\Pr(\phi_1) = \mu_E^{(t)} P_f \tag{3.21}$$

On this occasion, because the harvested energy from the RF signal is too little, we assume it is equivalent to zero packets of energy. The belief that the eavesdropper does not execute a jamming attack in the next time slot is given as

$$\mu_E^{(t+1)} = P_{\bar{J}\bar{J}} \tag{3.22}$$

Observation ϕ_2 The RF harvested energy is greater than the energy threshold, $e_{rf}^{(t)} \geq \epsilon$, which shows that the sensing result is correct, with probability

$$\Pr(\phi_2) = (1 - \mu_E^{(t)}) P_d \tag{3.23}$$

The belief that there will be no jamming attack in the next time slot is given as

$$\mu_E^{(t+1)} = P_{J\bar{J}} \tag{3.24}$$

Case 2

The sensing result shows that the eavesdropper is listening to the legitimate transmissions passively, with probability

$$\Pr\left(H_J^{(t)} = \bar{J} | \mu_E^{(t)}\right) = \mu_E^{(t)}(1 - P_f) + (1 - \mu_E^{(t)})(1 - P_d) \tag{3.25}$$

There are two possible situations:

(i) the source node does not have enough energy for data transmission, so it carries out RF energy harvesting and waits for more energy from the solar harvester (i.e. $e_{tr,S}^{(t)} = 0$);

(ii) the minimum required energy for data transmission is satisfied (e.g. $e_{r,S}^{(t)} > e_{tr1}$), and thus, the source uses a suitable number of energy packets to transmit data to the destination ($e_{tr,S}^{(t)} > 0$).

In the first situation, the source does not transmit data to the destination. Therefore, no reward is achieved: $R(s^{(t)}, e_{tr,S}^{(t)} = 0) = 0$. The remaining energy for the next time slot, $e_{r,S}^{(t+1)}$, is updated using Equation (3.19) with the transition probability given by Equation (3.20).

Observation ϕ_3 RF energy is less than the threshold, indicating that the sensing result is correct, with probability

$$\Pr(\phi_3) = \mu_E^{(t)}(1 - P_f) \tag{3.26}$$

The probability that the channel will be free of a jamming signal in the next time slot is updated by using Equation (3.22).

Observation ϕ_4 RF energy is greater than the threshold, which means that the jamming attack was wrongly determined, with probability

$$\Pr(\phi_4) = (1 - \mu_E^{(t)})(1 - P_d) \tag{3.27}$$

The probability that there will be no jamming attack in the next time slot is updated based on Equation (3.24).

Regarding the second situation, the source uses a finite number of energy packets to transmit data to the destination. The destination also transmits an interference signal against the eavesdropper. The remaining energy of the source for the next time slot is

$$e_{r,S}^{(t+1)} = \min\left(e_{r,S}^{(t)} + e_{h,S}^{(t)} - e_{tr,S}^{(t)}, E_{bat}\right) \tag{3.28}$$

with transition probability

$$\Pr\left(e_{r,S}^{(t+1)} | e_{r,S}^{(t)}, e_{tr,S}^{(t)} > 0\right) = \Pr\left(e_{h,S}^{(t)}\right) \tag{3.29}$$

The reward and the belief at the source are updated according to the acknowledgement (ACK) signal fed back from the destination after finishing the transmission, as described below.

Observation ϕ_5 The destination receives the information from the source, and then sends an ACK to the source confirming the transmission was successful. The probability that state $\bar{J}$ of the eavesdropper in the current time slot was identified correctly from the sensing result is given by

$$\Pr(\phi_5) = \mu_E^{(t)}(1 - P_f) \tag{3.30}$$

The belief for the next time slot is also given in Equation (3.22). Then, the reward for this case is the secrecy rate, which is

$$R\left(s^{(t)}, e_{tr,S}^{(t)} | ACK\right) = \log_2\left(\frac{1+\gamma_D^{(t)}}{1+\gamma_E^{(t)}}\right) \tag{3.31}$$

Observation ϕ_6 The destination cannot decode the information from the source, so it will send a negative ACK, which means misdetection about the presence of a jamming signal, with probability

$$\Pr(\phi_6) = (1 - \mu_E^{(t)})(1 - P_d) \tag{3.32}$$

Thus, no reward is achieved: $R(s^{(t)}, e_{tr,S}^{(t)} | \overline{ACK}) = 0$. The belief that the channel will be free of a jamming attack in the next time slot is calculated with Equation (3.24).

Based on those observations, the problem over the optimal transmit power in Equation (3.17) can be rewritten as follows:

$$e_{tr,S}^{*(t)} = \underset{e_{tr,S}^{(t)} \in \mathbb{A}}{\arg\max} \left\{ \begin{array}{l} \sum_{k=t}^{\infty} \beta^{k-t} \times \sum_{\phi_i \in e_{tr,S}^{(k)}} \Pr(\phi_i) \\ \times \sum_{e_{r,S}^{(k+1)}} \Pr\left(e_{r,S}^{(k+1)} | e_{r,S}^{(k)}, \phi_i\right) \\ \times R\left(s^{(k)}, e_{tr,S}^{(k)} | \phi_i\right) | s^{(t)} \end{array} \right\} \tag{3.33}$$

The final decision to maximize the secrecy rate can be found by solving Equation (3.33) using *value iteration*–based dynamic programming [42]. The flowchart of the proposed POMDP-based power decision scheme is given in Figure 3.4.

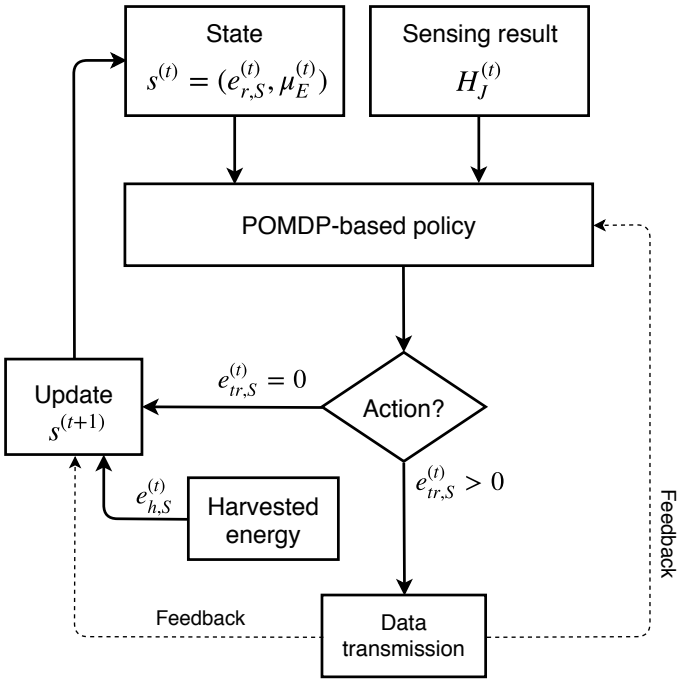

Figure 3.4: The flowchart of the proposed POMDP-based power decision scheme.

3.5 Actor–Critic Learning Framework for Energy-Efficient Data Transmission Against FD Eavesdropper

In the previous section, we presented a POMDP-based solution to the problem of power allocation on the assumption that the system has prior information about both the arrival of harvested energy and the jamming attack probabilities. However, it is not easy to know the environment's dynamics in advance, and more than that, the value iteration method requires complex and time-consuming computations. For this reason, in this section, we propose a model-free reinforcement learning framework, namely, an actor-critic approach, to solve the MDP problem. One of the advantages of this learning algorithm over the POMDP approach is that it does not require prior information about the environment's dynamics. This learning process also benefits from less formulation and computational effort. Using this method, the system learns the dynamics of the harvested energy by directly interacting with the environment. Therefore, we are going to present the implementation of a classical actor-critic learning framework to solve the MDP problem formulated in the previous section with the same state space $\mathbb{S}$ and action space $\mathbb{A}$.

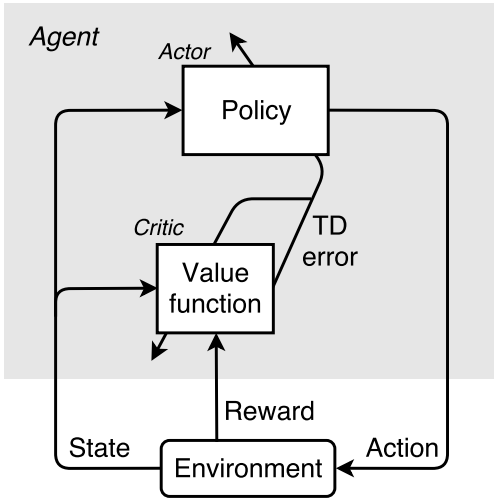

Figure 3.5: An actor-critic learning framework. TD: temporal difference.

Conventionally, an actor-critic architecture consists of three main elements [90]: an actor, which is a reference to a learning policy; a critic, which refers to a learning state-value function; and an environment, as illustrated in Figure 3.5. In the t^{th} time slot, the source deploys an action, $e_{tr,S}^{(t)} \in \mathbb{A}$, that is decided by the cluster head (i.e. the agent) based on the current state, $s^{(t)} \in \mathbb{S}$, and policy π, which is defined by using a softmax function as follows [1]:

$$\pi(s^{(t)}, e_{tr,S}^{(t)}) = \frac{e^{q(s^{(t)}, e_{tr,S}^{(t)})}}{\sum\limits_{a \in \mathbb{A}} e^{q(s^{(t)}, a)}} \tag{3.34}$$

where $q(s^{(t)}, e_{tr,S}^{(t)})$ is the value at time step t of the modifiable policy parameters of the actor, indicating the preference for selecting action $e_{tr,S}^{(t)}$ when in state $s^{(t)}$. This action causes the system to transit to a new state, $s^{(t+1)}$, with transition probability

$$\Pr(s' \in \mathbb{S} | s^{(t)}, e_{tr,S}^{(t)}) = \begin{cases} 1 & if\ s' = s^{(t+1)} \\ 0 & otherwise \end{cases} \tag{3.35}$$

and to return an immediate reward, $R(s^{(t)}, e_{tr,S}^{(t)})$, which can be calculated by using Equation (3.15).

The state-value function of the critic component is the total discounted reward

from the current time slot when the current state is s, which is given by [1]:

$$V(s) = R(s, \pi(s)) + \beta \sum_{s' \in \mathbb{S}} \Pr(s'|s, \pi(s)) V(s') \tag{3.36}$$

where β is the discount factor. The main goal of the actor-critic learning algorithm is to find the optimal policy, π^*, for the source to maximize this state-value function, and the problem in Equation (3.17) can be rewritten as

$$\pi^*(s) = \underset{e_{tr,S} \in \mathbb{A}}{\operatorname{argmax}} \left\{ R(s, e_{tr,S}) + \beta \sum_{s' \in \mathbb{S}} \Pr(s'|s, e_{tr,S}) V(s') \right\} \tag{3.37}$$

where $\Pr(s'|s, e_{tr,S})$ is the transition probability from state s to state s' after taking action $e_{tr,S}$, which is the transmit power of the source.

At the end of the time slot, the critic criticizes the action executed by the source based on a temporal difference (TD) error, which refers to the difference between the left side and the right side of the Bellman equation [1], as follows:

$$\delta^{(t)} = R(s^{(t)}, e_{tr,S}^{(t)}) + \beta V(s^{(t+1)}) - V(s^{(t)}) \tag{3.38}$$

It then uses this TD error to learn the state-value function associated with the system's current state, which is

$$V(s^{(t)}) \leftarrow V(s^{(t)}) + \alpha_c \cdot \delta^{(t)} \tag{3.39}$$

where α_c is a critic-positive step size. The actor also updates the policy by using the TD error as follows:

$$q(s^{(t)}, e_{tr,S}^{(t)}) \leftarrow q(s^{(t)}, e_{tr,S}^{(t)}) + \alpha_a \cdot \delta^{(t)} \tag{3.40}$$

where α_a is an actor-positive step size. The convergence rate of the algorithm is dependent on both α_c and α_a, which are differently designed based on various applications and empirical research.

The learning procedure of the actor-critic algorithm for the source to choose the optimal power for data transmission is detailed as follows. At the beginning of the t^{th} time slot, the source employs action $e_{tr,S}^{(t)}$ based on the sensing result, remaining energy $e_{r,S}^{(t)}$, and the stochastic policy, $\pi(s^{(t)})$. As described in the previous section, the source can operate in RF mode to harvest RF energy from radio signals and then waits for more harvested energy from the solar harvester (i.e. the transmit power is $e_{tr,S}^{(t)} = 0$), or it can use an appropriate number of energy packets ($e_{tr,S}^{(t)} > 0$) to transmit data to the destination.

The instant reward, $R(s^{(t)}, e_{tr,S}^{(t)})$, and the next state of the system, $s^{(t+1)}$, corresponding to each action are obtained based on the observations that are presented in the POMDP-based scheme. Since the actor-critic algorithm finds the decision policy from a practical learning process, it could converge to the locally optimal policy [91]. However, by using the actor-critic solution, we do not need to compute the state transition probabilities based on the probabilities of harvested energy to find the optimal action offline, as in the POMDP approach. It is worth noting that if we set the value of the discount factor to zero ($\beta = 0$) in both the POMDP-based approach and the actor-critic algorithm, the problem is equivalent to the myopic scenario, by which the agent only needs to maximize the secrecy rate in the current time slot without considering the effect of the present action on future rewards. The learning process of the proposed actor-critic algorithm for the power decision policy is summarized in Algorithm 1.

3.6 Simulation Results

3.6.1 Simulation setups

In this section, we present numerical simulation results to demonstrate the efficiency of the proposed POMDP and actor-critic schemes for energy-efficient data transmission against active eavesdroppers in WSNs. For the transmission links, all the wireless channels are modeled based on Rayleigh flat fading, and path-loss exponent ζ is set to 3.5. Regarding RF energy harvesting, we set the harvesting efficiency at $\rho = 0.5$. For the self-interference links at D and E, we assume the expectations of the link gains are normalized, and hence, the received signals depend mostly on the coefficient of self-interference. The legitimate transmit power at the source node ranges from $10mW$ to $50mW$. In our simulation, we fixed the locations of the source and the destination at coordinates $(0, 0)$ and $(50, 0)$, respectively (distances in meters). We verified the performance of the proposed schemes over 30,000 time slots, and the final results were obtained by averaging 10 independent runs. Unless otherwise presented, the main simulation parameters for the problem in this work are shown in Table 3.1. For analytical convenience, the jamming power of the destination (or of the eavesdropper) is fixed at $50mW$. Furthermore, when using actor-critic algorithm, we need to approximate the continuous-valued states and actions of the studied problem in this work with a finite number of discrete values. Therefore, the remaining energy and the

Algorithm 1 Actor-critic learning procedure for the transmit power decision policy at the source

Input: $\mathbb{S}$, $\mathbb{A}$, β, α_a, α_c, $P_{J\bar{J}}$, $P_{\bar{J}\bar{J}}$, H_J, P_d, P_f, P_D, P_E, E_s, E_{bat}, $E_{h,avg}$, E_{th}, T_{tr}, η.

Output: transmit power decision policy $\pi^*(s)$

1: Define a set of finite states, $\mathbb{S} = \{(e_{r,S}, \mu_E) : 0 < e_{r,S} \leq E_{bat}, 0 < \mu_E < 1\}$, in which each state represents the remaining energy of the source and the belief that the eavesdropper does not conduct jamming attack.

2: Define a set of finite actions, $\mathbb{A} = \{e_{tr,S} : 0 \leq e_{tr,S} \leq E_{tr,max}\}$, in which each action represents the energy consumption for data transmission, and $E_{tr,max}$ is the maximum transmit power of the source.

3: Determine the number of time slots for training, N.

4: Define a set of finite solar energy values and calculate the corresponding probabilities, with Equation (3.3) and Equation (3.4); then, generate an array of harvested energy values with size N.

5: Initialize state-value function $V(s)$ and policy $\pi(s, e_{tr,S})$, $\forall s \in \mathbb{S},\ \forall e_{tr,S} \in \mathbb{A}$.

6: **repeat**

7: At time step t, specify the current system state, $s^{(t)} = (e_{r,S}^{(t)}, \mu_E^{(t)})$.

8: Choose an action, $e_{tr,S}^{(t)} \in \mathbb{A}$, according to the initial policy when considering the sensing result and the remaining energy.

9: **if** $e_{tr,S}^{(t)} = 0$ **then**

10: perform RF harvesting

11: **else**

12: transmit data to the destination

13: **end if**

14: Calculate immediate reward $R(s^{(t)}, e_{tr,S}^{(t)})$ and update system state $s^{(t+1)}$ based on the observations.

15: Compute TD error $\delta^{(t)}$ with Equation (3.38).

16: Update state-value function $V(s^{(t)})$ with Equation (3.39).

17: Update the tendency to select an action, $q(s^{(t)}, e_{tr,S}^{(t)})$, and policy $\pi(s^{(t)}, e_{tr,S}^{(t)})$ with Equations (3.40) and (3.34).

18: **until** convergence or $t = N$.

19: Return final policy $\pi^*(s) = \underset{e_{tr,S} \in \mathbb{A}}{\arg\max}\{\pi(s, e_{tr,S})\}$.

Table 3.1: Simulation parameters

Notation	Description	Value
E_{bat}	Battery capacity of the source (packets)	100
E_s	Energy consumption for local spectrum sensing and sending the outcome to the cluster head (packet)	1
$E_{tr,max}$	Maximum transmit power of the source (packets)	50
$E_{h,avg}$	Average harvested energy (packets)	10
E_{th}	Energy threshold (packets)	3
$P_{J\bar{J}}, P_{\bar{J}J}$	Transition probabilities between states J and $\bar{J}$ of the jamming model	0.2
T_{tr}	Transmission duration (seconds)	0.2
η	Coefficient of residual self-interference	10^{-9}
σ_0^2	Background noise power (dBm)	-80
β	Discount factor	0.9
α_a, α_c	Actor and critic step-size parameters	$(0.1, 0.1)$
$\left(e_{r,S}^{(0)}, \mu_E^{(0)}\right)$	Initial state of the system	$(1, 0.1)$

transmit power at the source are quantized into $L_1 = 100$ and $L_2 = 10$ levels, respectively, in the simulation. We set the value of the desired global probability of detection at $P_d = 0.9$, and probability of false alarm at $P_f = 0.1$. It is worth noting that one energy packet is equivalent to $0.2mJ$. In the following simulation results, the performance of the proposed schemes is compared with the myopic scheme, under which the decision is only made for the current time slot to maximize the secrecy rate, as studied in [92] and in [93].

3.6.2 Performance evaluation

We first inspected the convergence rate of the actor-critic algorithm during the learning process with different step-size parameters, α_a and α_c, based on the reward (i.e. secrecy rate) computed every 1000 time slots. In this simulation, the position of the eavesdropper is fixed at coordinate $(0, -75)$. It is worth noting that the convergence condition of the proposed actor-critic algorithm is the convergence of the reward. During the training process, we regularly computed the average reward after every batch of a thousand training slots and then calculated the difference between two adjacent values, ΔR. In this work, the convergence condition is defined as $|\Delta R| < 0.005$.

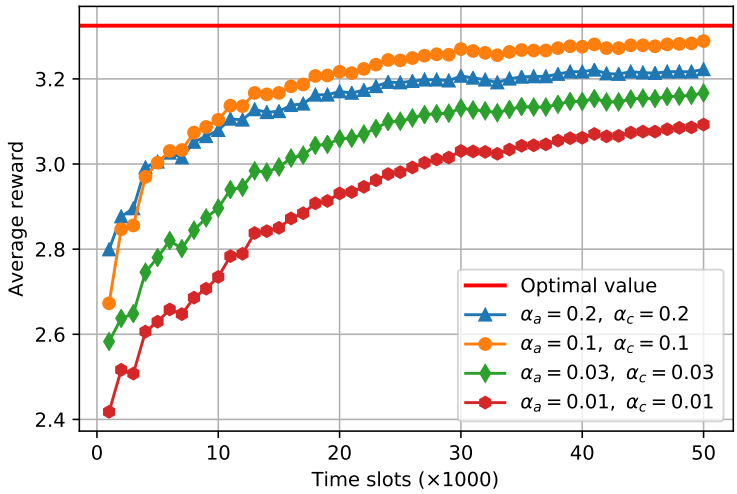

Figure 3.6: Convergence of the proposed actor-critic algorithm with different values of step-size parameters.

As can be observed from Figure 3.6, the average reward after each iteration of 1000 time slots increases significantly in the first 10,000 time slots of the learning process, and then, the algorithm converges to the optimal policy, depending on the values of the actor and critic learning rates. To be specific, with step-size values smaller than 0.1, the larger step-size parameters provide the agent with better rewards and faster convergence speed. Furthermore, the actor-critic algorithm tends to converge to the optimal policy after 30,000 time slots, which almost reaches the optimal value given by the POMDP-based solution. From the figure, we can see that to keep increasing the step-size parameters does not provide the system with better rewards. Instead, the algorithm may converge to a locally optimal policy due to over fitting. Therefore, we choose the actor and critic step sizes as $\alpha_a = 0.1$ and $\alpha_c = 0.1$ for the actor-critic algorithm in other simulations.

In order to examine the system behavior for different locations of the active eavesdropper, we present the legitimate user's secrecy rate and the wiretap rate of the eavesdropper when the eavesdropper moves along a straight line from $(0, -75)$ to $(50, -75)$, as depicted in Figure 3.7. It is clearly shown that when the eavesdropper moves farther from the source node, the wiretap rate drops significantly while the secrecy rate increases. The reason is that when the eavesdropper moves from left to right, the source–eavesdropper distance increases while the eavesdropper–destination decreases. Consequently, the SINR at the eavesdropper drops significantly. The figure shows that the secrecy performance of the

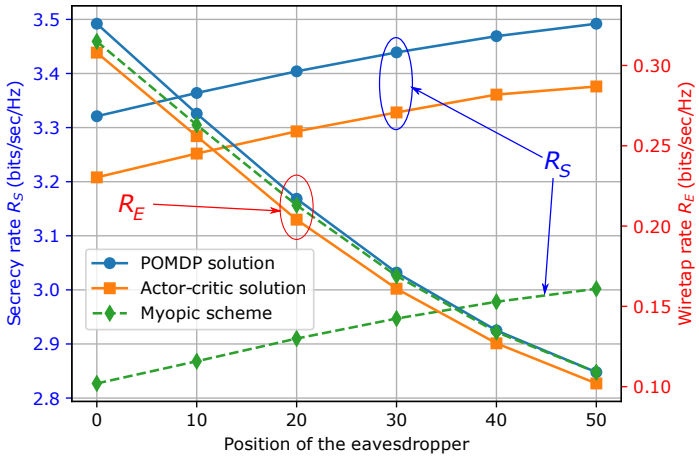

Figure 3.7: The legitimate user's secrecy rate and the eavesdropper's wiretap rate with an active eavesdropper locating itself at different positions.

proposed schemes is better than the myopic scheme. Meanwhile, the difference among the wiretap rates of the power allocation schemes is insignificant. This is because, when using the proposed schemes, the source efficiently allocates the transmit power in each time slot based on the information about the arrival of harvested energy and the jamming activity. In the following simulations, only the numerical results for one location of the eavesdropper, at coordinate $(0, -75)$, are shown for the sake of simplicity.

In Figure 3.8 and Figure 3.9, we illustrate the effect of the source's battery capacity on the performance of the proposed solutions, compared with the myopic scheme, when the average harvested energy is $E_{h,avg} = 10$. As can be seen from the figures, when the battery capacity, E_{bat}, increases, source S can store more harvested energy in its battery, and thus, it can transmit data with higher power, which results in a higher legitimate transmission rate, R_D. In addition, the transmission rates of the proposed algorithm dominate that of the myopic scheme. The reason is that, in the myopic scheme, if there is no jamming signal (based on the sensing result), S will use most of its energy for the transmission process in the current time slot to maximize the transmission rate, without considering the effect of this action on future rewards. However, due to the limitation of the available harvested energy, using too much energy in a time slot can cause the source to become inactive in many future time slots, which leads to lower transmission rates. On the other hand, the two proposed solutions can guarantee the long-term performance of the system relying on effective energy allocation for data transmission in each time slot.

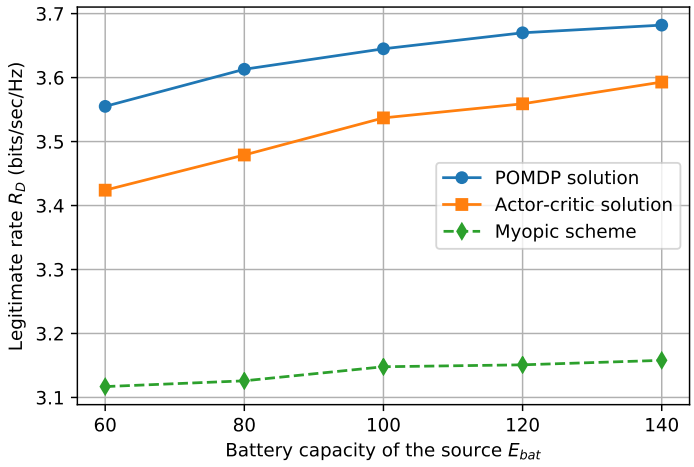

Figure 3.8: The legitimate user's average transmission rate according to battery capacity of the source.

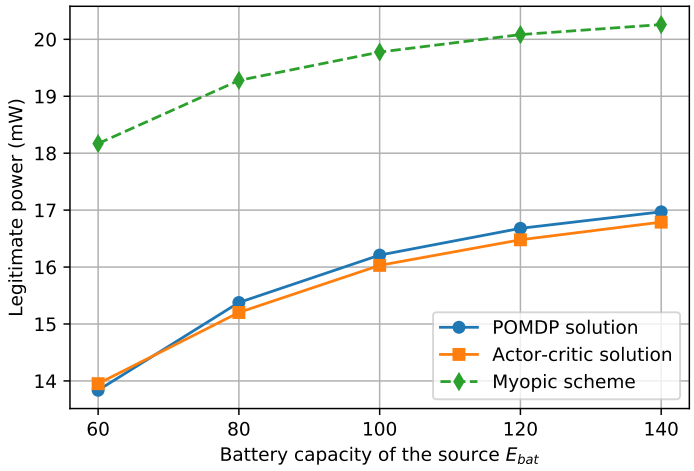

Figure 3.9: The legitimate user's average transmit power according to battery capacity of the source.

Similarly, in Figure 3.10 and Figure 3.11, we compare the performance of the power allocation schemes under the effect of the harvested energy. As seen in the figures, a larger amount of harvested energy provides the source with higher secrecy rates. Obviously, if S can harvest more solar energy, it has more chances to operate in transmission mode, and it can also transmit more data to the destination using higher transmit power. Furthermore, the results of the POMDP and the actor-critic approaches are better than the remaining scheme. To explain this, in the two proposed schemes, the source allocates power for

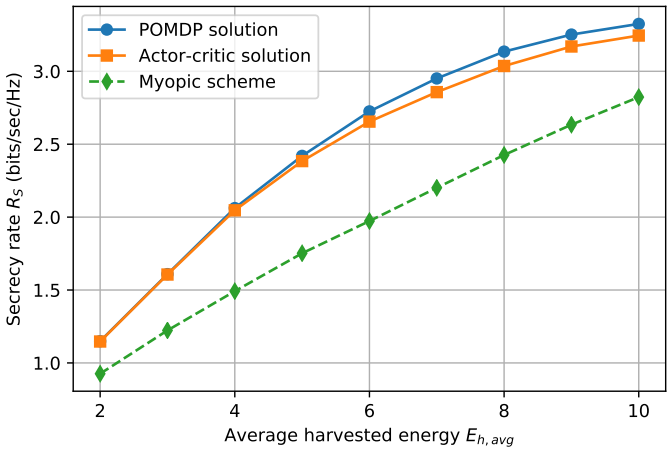

Figure 3.10: Secrecy rate according to harvested energy.

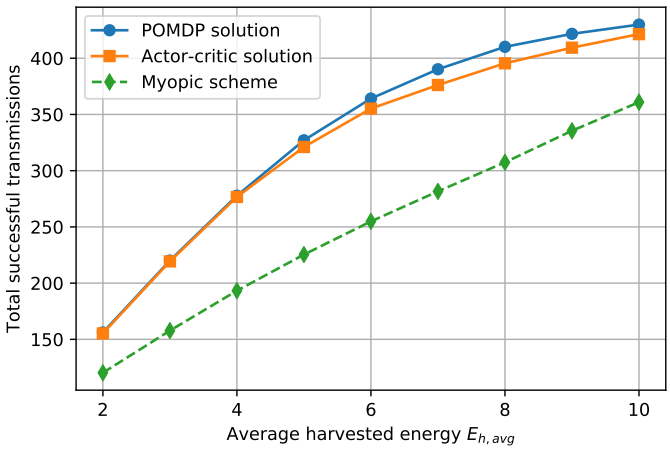

Figure 3.11: Total successful transmissions according to harvested energy. The result is obtained from 1000 time slots.

the transmission process with awareness of the arrival of harvested energy during each time slot and the status of the jamming activity. More specifically, in the POMDP-based algorithm, the information about the arrival of harvested energy and about the model of the jamming attack are assumed to be available; hence, the cluster head can compute all possible situations and their corresponding probabilities to find the optimal policy beforehand. As a result, the next state of the system is predictable, so the source can efficiently distribute the energy over future time slots. Meanwhile, when using the actor-critic algorithm, the source needs to regularly interact with the environment so it can learn the dynamics of

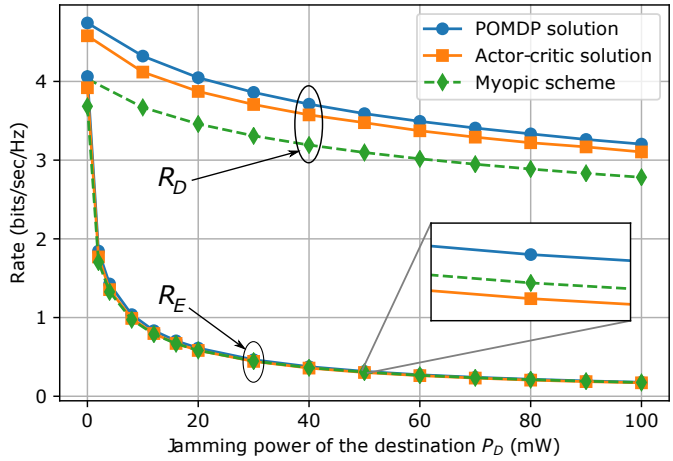

Figure 3.12: The legitimate user's transmission rate and the wiretap rate according to the jamming power of the destination.

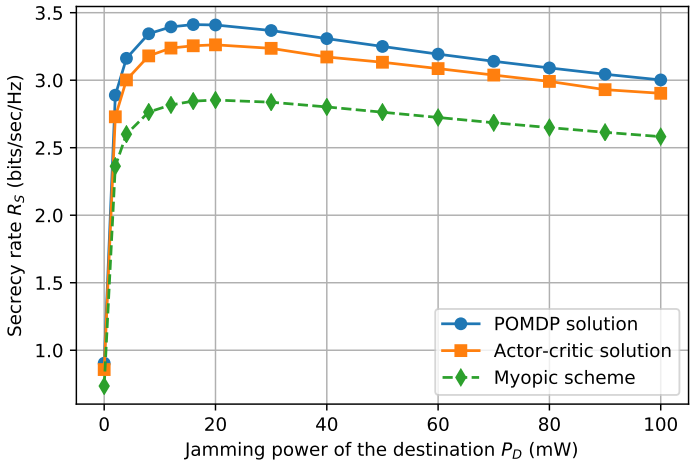

Figure 3.13: The secrecy rate according to the jamming power of the destination.

the harvested energy, which can result in a locally optimal policy. Therefore, the legitimate rate and the number of successful transmissions under the actor-critic scheme are lower than those of the POMDP algorithm. One transmission in a time slot is considered successful if there is no jamming attack in that time slot.

Figures 3.12 and 3.13 show the effect on the legitimate transmission rate, the wiretap rate, and the secrecy rate of the transmit power that the destination uses to jam the eavesdropper. As we can see from Figure 3.12, both the legitimate transmission rate and the

wiretap rate decrease as the jamming power of the destination increases. Specifically, if the destination does not send an interference signal against the eavesdropper (i.e. $P_D = 0$), the wiretap rate is remarkably high. In addition, when the jamming power of the destination rises from $0mW$ to $20mW$, the wiretap rate declines significantly while the secrecy rate grows notably. This is because the wiretap rate decreases much faster than the legitimate transmission rate. When the destination's transmit power goes above $20mW$, the effect of the self-interference on the legitimate transmission becomes more apparent, and thus, the secrecy rate starts to reduce gradually. More importantly, the secrecy rates in our proposed solutions, which optimize the energy allocation for use in future time slots, are better than under the myopic scheme. This is because the myopic scheme tries to maximize the transmission rate by using more transmit power, which drains the source of energy to be used in the future; hence, it has to stay inactive in the next few time slots.

We further investigate the joint impact of the jamming power of both the destination and the eavesdropper on the system reward. Figure 3.14 shows the estimate of the system reward when the values of P_D and P_E are changed. In general, the average cumulative rewards of the system decrease with the increase of P_D or P_E. Furthermore, when we increase the destination's jamming power from $P_D = 10mW$ to $P_D = 100mW$, the system rewards increase at first but decrease if P_D keeps increasing. The reason is that higher P_D causes significantly low SINR at the eavesdropper, and subsequently, increases the secrecy rate. However, when P_D is too high, the effect of the self-interference on the received signal at the destination cannot be neglected, which leads to a significantly low transmission rate at the destination. The system rewards also fall if the eavesdropper arises its jamming power to interfere with the destination. The reason is that the source might need to use more power for data transmission to increase the secrecy rate, which might cause it to stay inactive in many time steps due to the low energy level in its battery. From the figure, we can see that there exists an optimum value for the destination's jamming power, which is located around the line $P_D = 20mW$. At this point, the two proposed algorithms provide the best rewards.

Figures 3.15, 3.16, and 3.17 demonstrate the security performance of the proposed schemes with different coefficients of self-interference. In Figure 3.15, we show the transmit power allocation at the source for different power allocation approaches. We can see that the transmit power allocation under the myopic scheme does not change as the coefficient of self-interference varies. Obviously, this scheme makes decisions on power allocation without

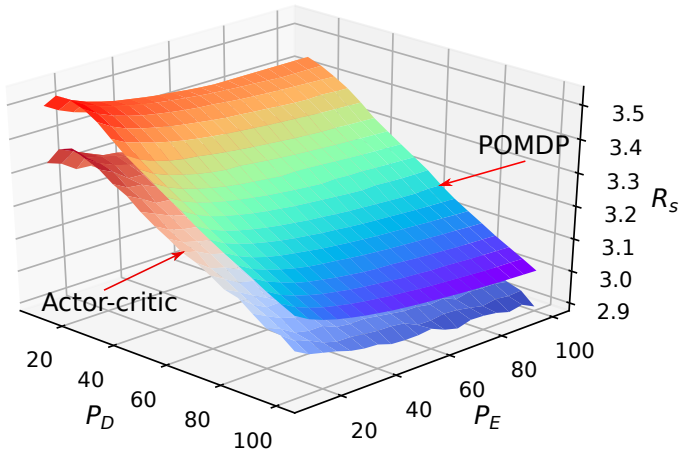

Figure 3.14: The secrecy rate under different values of P_D and P_E.

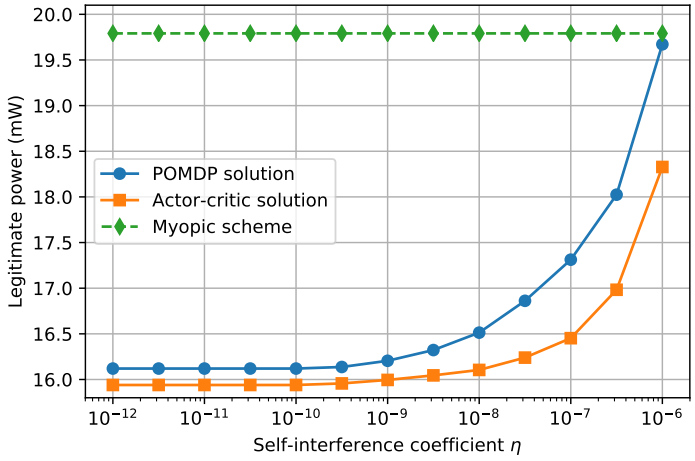

Figure 3.15: The legitimate user's transmit power according to the coefficient of self-interference.

considering the effect of self-interference. Therefore, when using the myopic scheme, the source allocates the maximum transmit power to the transmission process. As a consequence, the secrecy rate obtained with this scheme is lower than the proposed solutions, as depicted in Figure 3.16.

On the other hand, with the POMDP-based and the actor-critic solutions, the transmit power does not change much when the self-interference coefficient is sufficiently small, because the impact of self-interference in these cases is negligible. However, when the coefficient is large enough, for which the influence of self-interference on the destination,

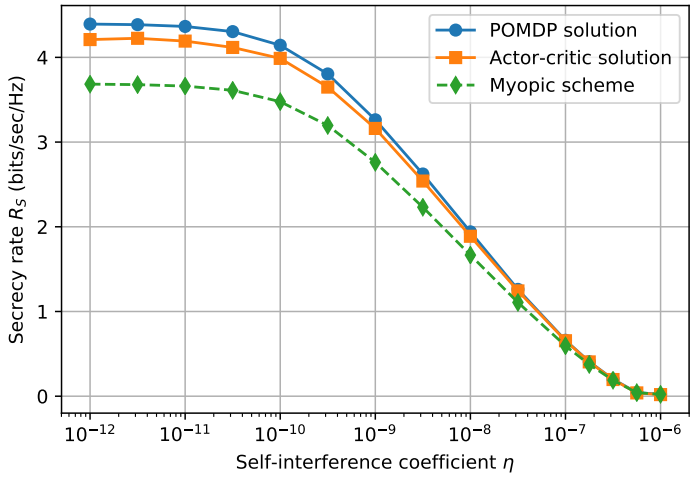

Figure 3.16: The secrecy rate according to the coefficient of self-interference.

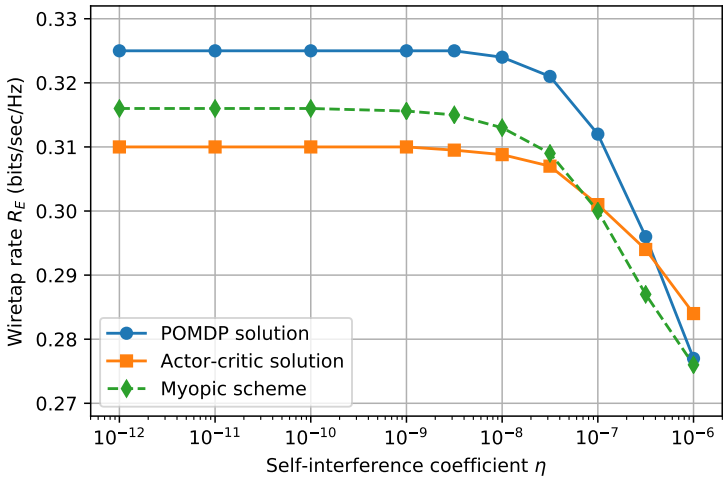

Figure 3.17: The wiretap rate according to the coefficient of self-interference.

as well as the eavesdropper, becomes a dominant factor, the transmit power of the source under the two proposed solutions starts to increase remarkably. The aim of the increment in the transmit power is to improve the secrecy rate, which is reduced sharply as the self-interference coefficient increases. A similar situation can be observed for the wiretap rate at the eavesdropper. The reason is that, when the coefficient of self-interference is small, the wiretap rates at different coefficients are almost the same. Nevertheless, the growth of the attenuation factor (beyond the value 10^{-8}) makes the wiretap rate much smaller.

Consequently, we can verify that the proposed POMDP-based solution and the actor-critic algorithm can ensure energy-efficient data transmissions between the source (i.e. the sensor node) and the destination (i.e. the cluster head) against a full-duplex active eavesdropper in wireless sensor networks.

3.7 Conclusion

In this chapter, we present two energy-efficient power allocation schemes for data transmission against a full-duplex active eavesdropper in a cognitive-aided wireless sensor network. In this network, the sensor nodes can harvest energy from both non-RF and RF resources, and the cluster head has the ability to use FD communications, which allows it to transmit and receive signals at the same time. The eavesdropper can opportunistically send a jamming signal to the destination while eavesdropping, and the cluster head can interfere with this process by sending artificial noise against the eavesdropper. To secure data in the presence of an active eavesdropper, the source first performs spectrum sensing to determine whether the eavesdropper is acting passively or not, and a global decision about the jamming activity is made by the cluster head. Along with the jamming state decision, in each time slot, the cluster head also sends the power allocation policy to the source on the control channel. Based on this policy, the source effectively allocates power for data transmissions to maximize the long-term secrecy rate of the system under the constraint of harvested energy. The simulation results show that our proposed solutions can effectively enhance data secrecy and energy utilization when self-interference is sufficiently suppressed.

Chapter 4

Actor-Critic Deep Learning for Efficient User Association and Bandwidth Allocation in Dense Mobile Networks with Green Base Stations

4.1 Introduction

The tremendous growth in mobile devices and the rapidly increasing demands for multimedia services have led to an issue of efficient resource allocation in wireless networks due to the scarce availability of radio spectrum. At present, mobile networks are expected to be deployed with extensively low-cost and low-power small-cell base stations (SBSs) to enhance the overall performance of the system [94]. In wireless communications, different real-time applications might use different encoding schemes according to their desired quality, and they thus generate different bandwidth requirements [95]. In small-cell wireless networks where the spectrum band is a scarce resource, spectrum efficiency become extremely challenging due to the intensive characteristics of dense deployment and the stochastic property of mobile users [96]. As a consequence, many solutions have been proposed to optimize bandwidth allocation based on the different criteria of the network [97, 98].

Along with the challenges in spectrum management, energy-efficiency in green communications has become one of the major concerns for network management, especially in small-cell networks that are powered by ambient energy sources [99]. One of the prospective ways to enhance the self-sustainability of such a network is to equip the SBSs with rechargeable batteries integrated with energy harvesting devices. This method can ensure energy autonomy in the network by utilizing renewable energy to regularly recharge the limited-capacity batteries of the SBSs [100]. Among various types of renewable energy, solar power, harvested directly from sunlight, is considered the most common and effective energy resource [101]. However, the capacity to generate solar power is highly dependent on the environmental conditions, and may vary with time. Therefore, it is essential to attain an efficient energy management policy to improve long-term network performance.

Many studies have addressed the problem of energy-efficient resource allocation in small-cell networks. Conventionally, research on energy-efficient resource allocation in mobile networks usually aims to balance the traffic load among the base stations in order to enhance spectrum efficiency and energy conservation [102–104]. Xie *et al.* [105] studied the energy-efficiency aspect of spectrum sharing and power allocation in heterogeneous cognitive radio networks with femtocells. They proposed a gradient-based iteration algorithm to obtain the solution to the energy-efficient resource allocation problem. In [106], the authors investigated a joint service-pricing and bandwidth-allocation problem at the operator level for energy efficiency in heterogeneous network deployment (e.g., composed of macrocells, microcells, and femtocells). However, most of the existing work on resource allocation in wireless networks assumes that the variations in traffic load are predictable, which is often not true in wireless networks due to user mobility.

Since accurate information about traffic load and the arrival of harvested energy is sometimes unavailable, researchers usually formulate stochastic optimization problems in mobile networks as the framework of a Markov decision process (MDP) [80]. Afterwards, the solution to the formulated MDP can be attained by making use of reinforcement learning (RL) approaches [107]. In reinforcement learning, the agent does not need to know the environment's dynamics in advance, and learns the optimal decision policy through interactions with the environment [1]. For example, Wei *et al.* [82] proposed an actor-critic algorithm to find an optimal policy for user scheduling and resource allocation in HetNets powered by hybrid energy for the purpose of maximizing the energy efficiency of the network. However, for conventional RL methods, it is a big challenge to solve problems

with large state and action spaces. For this reason, deep reinforcement learning (DRL) techniques have recently become more popular in solving optimization problems in wireless communications. In DRL, deep neural networks (DNNs) serve as function approximators (e.g., of the value function), and are used to learn the policy [108]. Yu *et al.* [109] proposed a DRL algorithm to improve the total throughput of multiple networks by sharing time slots among co-existing wireless networks.

To the best of our knowledge, there is little research using DRL methods for resource allocation in wireless networks with energy-harvesting base stations. Therefore, we propose an actor-critic deep learning framework for efficient resource allocation in dense mobile networks, in which solar-power harvesters supply energy to the base stations. This work aims to find the optimal user-association and bandwidth-allocation policy for downlink data services in order to maximize long-term network performance. In particular, the contributions of this work are as follows.

- A joint user-association and bandwidth-allocation problem for downlink data transmission is proposed, considering the stochastic arrival of data requests and harvested energy in the network. The base stations share the total system bandwidth, and utilize the harvested energy for data transmission. The system reward is the overall satisfaction ratio of the users in the network, and the goal is to maximize the accumulated reward in the long run.

- The optimization problem is then reformulated as the framework of an MDP. We attain the solution to the formulated MDP by employing an actor-critic learning framework, which is a trial-and-error learning algorithm. Furthermore, we use DNNs to approximate the policy function and the value function in the actor and the critic, respectively. Specifically, the agent in our proposed algorithm evaluates the dynamics of harvested energy and data requests in the network through interactions with the environment to find the optimal decision-making policy, and thus, to maximize the system reward.

The remainder of this chapter is structured as follows. Section 4.2 introduces the system model and the problem formulation. Section 4.3 presents the actor-critic deep learning algorithm to find the optimal user-association and bandwidth-allocation policy. Section 4.4 presents a performance analysis of our proposed scheme through numerical simulation results. Section 4.5 concludes the chapter.

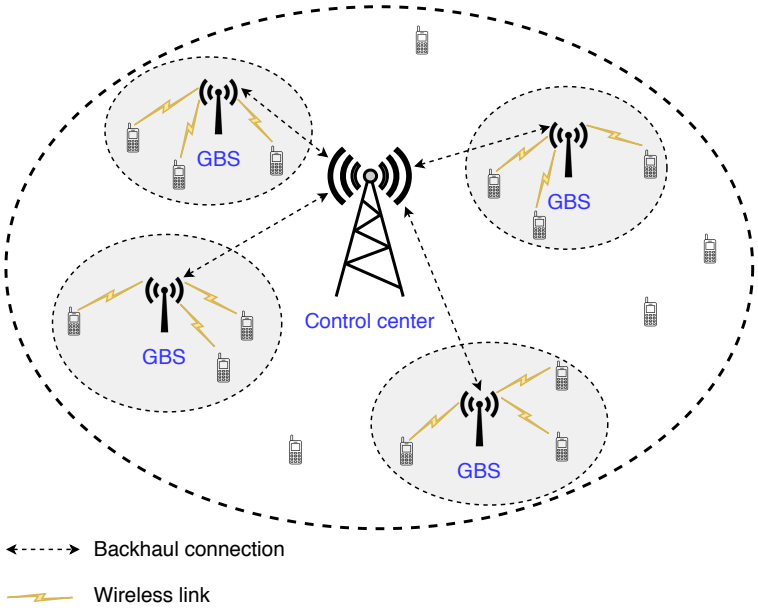

Figure 4.1: A mobile wireless network with energy-harvesting base stations.

4.2 System Model and Problem Formulation

4.2.1 System model

We consider a mobile wireless network consisting of K green base stations (GBSs) integrated with energy-harvesting components (i.e., solar panels) and U mobile users that are requesting data services from the network, as shown in Figure 4.1. In this network, the GBSs are densely deployed within a macro cell and share the same spectrum band. We assume that the GBSs communicate with a control center via backhaul connections with perfect information exchange and that the controller can make intelligent resource allocation decisions for the whole network in a centralized way. We focus on user association and bandwidth allocation on network downlink. We denote the set of base stations and the set of users as $\mathcal{K}$ and $\mathcal{U}$, respectively. The system is assumed to operate over a time-slotted basis, in which a time slot is denoted by t. For the sake of notation simplicity, the control center is called the controller, and we use the words *base station* (BS) to represent the green base stations. We also use the terms *time slot* and *time step* interchangeably.

The operation of the system proceeds as follows. At a given time step, each user might issue data requests to the base stations with probability p_r. Based on the available

system bandwidth and the remaining energy of each BS, the controller first decides whether to associate a user with a base station or not, then allocates bandwidth to the user for the data service. The purpose of user association is to offload the traffic from the macrocell to the small cells, and thus, enhance network performance in terms of high data rates and low power consumption. We assume that each user can receive data service from only one base station in a time slot. Let $x_{k,j}(t) \in \{0, 1\}$ denote the association variable between BS $k \in \mathcal{K}$ and user $j \in \mathcal{U}$ in time slot t (i.e., $x_{k,j}(t) = 1$ if user j is associated with BS k; otherwise, $x_{k,j}(t) = 0$). Hence, the number of users associated with BS k is given by

$$u_k(t) = \sum_{j=1}^{U} x_{k,j}(t) \tag{4.1}$$

Regarding bandwidth allocation decisions, we define the total system bandwidth as the number of channels that are available for data transmission in the context of this work, as discussed in [110]. The purpose of bandwidth allocation is to determine the optimal bandwidth that should be allocated to each base station to maximize the number of users that can receive service from the network. For the sake of mathematical simplicity, we assume that the BSs need one channel with a fixed transmission rate to establish a connection and provide service to each user. Therefore, the total bandwidth required at base station k in time slot t is also denoted by $u_k(t)$. Due to the limitation in frequency resources, we further assume that the base stations schedule the allocated bandwidth for the associated users by using a simple first-come-first-served mechanism. Let $y_j(t) \in \{0, 1\}$ denote the bandwidth allocation variable for user j in time slot t (i.e., if user j is allocated bandwidth for data service, $y_j(t) = 1$; otherwise, $y_j(t) = 0$). Hence the amount of bandwidth allocated to the users at BS k is given by

$$b_k(t) = \sum_{j=1}^{U} x_{k,j}(t) \times y_j(t) \tag{4.2}$$

where $0 \leq b_k(t) \leq B_{max}$, and B_{max} denotes the total system bandwidth. We define the service time of each request as the number of time slots required to complete a transmission using one channel, and this service time is assumed to follow an exponential distribution with mean μ_s. We further consider the cell-sojourn time (the number of time slots a user stays in a cell) as the deadline for each request, and we also approximate this deadline using an exponential distribution with mean μ_d. It is important to note that the deadline for each

request will be reduced by 1 (slot) after every time step, and that the base station will not retransmmit if the request's deadline reaches zero.

4.2.2 Problem formulation

Our target is to maximize the long-term performance of the system in terms of customer satisfaction while ensuring energy conservation. We focus on evaluating the impact of the amount of allocated bandwidth on the satisfaction degree. Furthermore, we assume that the controller can perform perfect resource timing and scheduling, hence the problem is formulated without considering the effects of time delays (e.g., channel access delay and queuing delay) and packet losses in each time slot. For this purpose, we first define the satisfaction of the users at base station k in time slot t as the ratio of the total allocated bandwidth to the total required bandwidth, as follows:

$$SR_k(t) = \begin{cases} \dfrac{b_k(t)}{u_k(t)} & \text{if } u_k(t) > 0 \\ 0 & \text{if } u_k(t) = 0 \end{cases} \tag{4.3}$$

where the condition ($u_k(t) = 0$) indicates that no user is associated with base station k, and thus, no user is served by the base station. The system reward at time t is the average satisfaction ratio in the network, which can be computed as

$$R(t) = \frac{1}{K} \sum_{k=1}^{K} SR_k(t) \tag{4.4}$$

where K is the number of base stations in the network. Eventually, the controller needs to find an effective policy for user association and bandwidth allocation to maximize the expected long-term reward within the constraints of harvested energy and system bandwidth.

In our problem, the BSs are powered solely by solar energy-harvesting devices. Each BS is equipped with a rechargeable battery with finite capacity E_{max} to store the harvested energy. We denote the number of energy packets that BS k can harvest during time slot t as

$$e_k^h(t) \in \left\{ e_1^h, e_2^h, ..., e_\zeta^h \right\} \tag{4.5}$$

In this work, $e_k^h(t)$ is assumed to follow a Poisson point process with mean λ_e. Therefore, the probability mass function of $e_k^h(t)$ can be given as

$$p^h(i) = \Pr\left[e_k^h(t) = e_i^h \right] = \frac{e^{-\lambda_e} (\lambda_e)^i}{i!}, \; i = 1, 2, \ldots, \zeta \tag{4.6}$$

It is worth noting that the information about the current energy level in each BS is reported to the controller at the beginning of each time slot via the control channel. In this work, the BSs use finite energy packets, denoted by e, to transmit data to the users using one channel. We further denote the current energy level in the battery of BS k as $e_k(t)$, which shows the temporal capability of transmitting data.

Due to the limitation in system bandwidth and the battery capacity of the base stations, the controller should make intelligent decisions under the following constraints. Unless otherwise denoted, the summations over base station index k and user index j extend over all of $\mathcal{K}$ and all of $\mathcal{U}$, respectively.

1. User-association constraint: one user can be associated with only one BS at a time. Thus, we have
$$\sum_{k} x_{k,j}(t) = 1 \tag{4.7}$$

2. Bandwidth constraint: given a temporary system bandwidth, $B(t) \in [0, B_{max}]$, the bandwidth allocated to the base stations should satisfy
$$\sum_{j} y_j(t) \leq B(t) \tag{4.8}$$

3. Energy constraint: the total energy consumption at BS k for data transmission using the allocated bandwidth should not exceed the current energy level of the BS, as follows:
$$e \times \sum_{j} x_{k,j}(t) \times y_j(t) \leq e_k(t) \tag{4.9}$$

As a consequence, the satisfaction maximization problem in this work can be written as:

$$\begin{aligned} \max_{\boldsymbol{X},\boldsymbol{Y}} \quad & \sum_{t=0}^{\infty} \eta^t R(t) \\ s.t. \quad & \textstyle\sum_k x_{k,j}(t) = 1 \\ & \textstyle\sum_j y_j(t) \leq B(t) \\ & \textstyle\sum_j x_{k,j}(t) \times y_j(t) \leq \frac{e_k(t)}{e} \\ & x_{k,j}(t) \in \{0,1\},\ \forall k \in \mathcal{K} \text{ and } \forall j \in \mathcal{U} \\ & y_j(t) \in \{0,1\},\ \forall j \in \mathcal{U} \end{aligned} \tag{4.10}$$

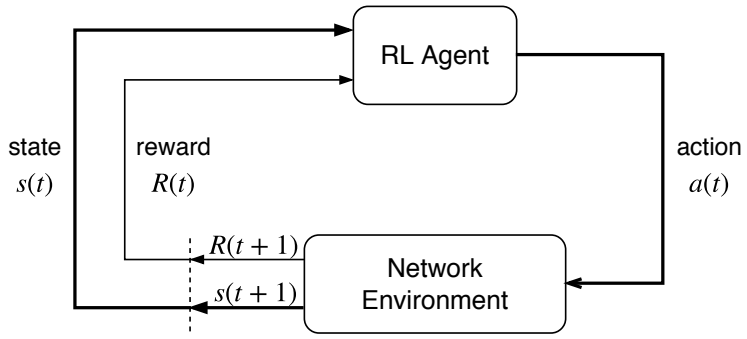

Figure 4.2: The agent-environment interaction in a decision-making process. Source: Adapted from [1].

where $\boldsymbol{X} \triangleq [x_{k,j}(t)]_{K\times U}$ is the user association matrix, $\boldsymbol{Y} \triangleq [y_j(t)]_{1\times U}$ is the bandwidth allocation vector, $\eta \in [0, 1]$ is a discount factor that maps the future rewards to the current time step, and $\sum_{t=0}^{\infty} \eta^t R(t)$ is the cumulative discounted rewards of the system. Since the arrival of traffic requests and harvested energy packets is random, and while the system has limited bandwidth resources, and each BS has a finite-capacity battery, the controller needs to deploy an actor-critic deep learning algorithm to estimate the variations in the network environment. As a consequence, the controller can find the optimal user-association and bandwidth-allocation scheme to maximize the network performance in the long run.

4.3 Actor-Critic Deep Learning Framework

We reformulated the stochastic optimization problem of joint user association and bandwidth allocation in mobile networks as the framework of a Markov decision process. Since the arrivals of harvested energy and data requests are unknown, we employ a model-free RL framework to find the solution to the formulated problem, in which the RL agent learns the optimal decision-making policy via trial-and-error interactions with the environment.

4.3.1 Markov decision process

A standard RL model consists of an agent periodically interacting with an environment over time, as depicted in Figure 4.2. In an RL system, the agent learns how to map environment states to a suitable action model, formally called a policy, through a

trial-and-error learning process to maximize the accumulated sum of rewards. First, we define the state space, the action space, and the reward function of the system. The state space of BS k is denoted by

$$\mathcal{S}_k = \{(e_k); e_k \in \{0, 1, \ldots, E_{max}\}\} \tag{4.11}$$

where e_k represents the number of energy packets in the battery of the base station. The state space of the system bandwidth is denoted by

$$\mathcal{S}^B = \{(B); B \in \{0, 1, \ldots, B_{max}\}\} \tag{4.12}$$

where B is the available system bandwidth. Hence, the state space of the environment is determined by the Cartesian product of $\mathcal{S}^B$ and all $\mathcal{S}_k$, as follows:

$$\mathcal{S} = \mathcal{S}^B \times \prod_{k=1}^{K} \mathcal{S}_k \tag{4.13}$$

The decision-making process proceeds as follows. At the beginning of time slot t, the agent observes state $s(t)$ in state space $\mathcal{S}$ about the environment, and then chooses action $a(t)$ in action space $\mathcal{A}$ following a stochastic policy.

In our work, the network agent (i.e., the controller) decides whether to associate a user with a base station and whether to allocate a radio channel to the user in each time slot. Therefore, the action $a(t)$ is set as

$$a(t) = \{x_{k,j}(t), y_j(t)\}_{\forall k \in \mathcal{K}, \forall j \in \mathcal{U}} \tag{4.14}$$

where $x_{k,j}(t)$ and $y_j(t)$ are the user-association variable and the bandwidth-allocation variable, respectively. In particular, if no user is associated with BS k in time slot t (i.e., $\sum_j x_{k,j}(t) = 0$), the base station has to stay inactive, and it waits for more harvested energy in the current time slot. Otherwise, the base station will transmit data to the users using the allocated channels. At the end of the time slot, the base station reports the information about the number of satisfied users and the current energy level in its battery to the controller for network management.

Thereafter, the environment feeds back the immediate reward, $R(t)$, which is defined in Equation (4.4), to the agent, and transforms to a new state, $s(t+1)$. The environment state in the next time slot, $s(t+1) = \{B(t+1), e_k(t+1)\}_{\forall k \in \mathcal{K}}$, is updated as follows. The available system bandwidth that could be used in the next time slot is given by

$$B(t+1) = B(t) - \sum_j y_j(t) + B_{re}(t) \tag{4.15}$$

where $B_{re}(t)$ denotes the bandwidth that is released at the end of the t^{th} time slot. Meanwhile, the remaining energy in each BS is updated based on the following observations.

Observation 1

BS k serves no user in time step t (i.e., $SR_k(t) = 0$), and thus, the base station stays idle and waits for harvested energy from solar panels. The reason for this might be due to the limitation in either the battery's energy or the system bandwidth. The energy level in the battery of BS k for the next time slot is updated as

$$e_k(t+1) = \min\left(e_k(t) + e_k^h(t), E_{max}\right) \tag{4.16}$$

Observation 2

If BS k receives a positive amount of bandwidth for data transmission (i.e., $b_k(t) > 0$), the remaining energy in this base station for the next time slot will be

$$e_k(t+1) = \min\left(e_k(t) + e_k^h(t) - e \times b_k(t), E_{max}\right) \tag{4.17}$$

where $e \times b_k(t)$ is the total energy consumption at BS k for transmitting data to the users.

Consequently, we aim to find the optimal action at the current time step to maximize the expected value of accumulated rewards, which is usually called the state-value function, following a given policy. In our work, the state-value function is denoted as [1]

$$V^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \eta^t R(t) | s(0) = s, \pi\right] \tag{4.18}$$

where $E[.]$ denotes the expectation, and π is the stochastic policy that maps the environment state s to the probability of taking action a, $\pi(a|s) = \Pr(a(t) = a|s(t) = s)$. The final purpose is to find the optimal policy, π^*, to maximize the discounted value function starting from state $s \in \mathcal{S}$, which satisfies the Bellman equation as follows [111]:

$$\pi^*(s) = \arg\max_{\pi} \{V^{\pi}(s)\} \tag{4.19}$$

Conventionally, we can use value iteration–based dynamic programming [1] to find the optimal decision policy for the MDP problem. However, this method heavily depends on prior information about the environment's variations, which is usually unknown in practice. Furthermore, it is challenging to directly compute state values using the Bellman equation

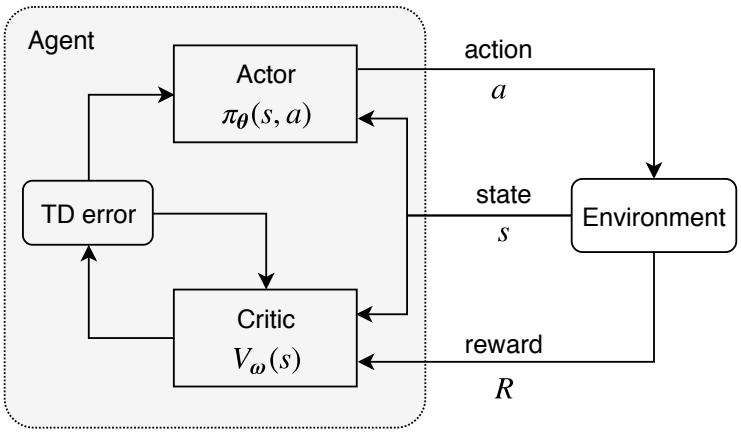

Figure 4.3: The structure of the actor-critic learning framework. TD error: temporal-difference error.

in high-dimensional state space and action space. Therefore, we employ an actor-critic algorithm to solve the MDP problem without requiring information about the environment's dynamics in advance. We additionally use deep neural networks to model the policy function and the value function of the actor-critic agent so that the algorithm can work effectively with large state and action spaces. Therefore, the proposed method is called the actor-critic deep learning (ACDL) algorithm.

4.3.2 The actor-critic deep learning framework for user association and bandwidth allocation in dense mobile networks

The actor-critic algorithm proposed in [112] aims to combine the strength of policy-based methods [113] and value-based methods [114]. A typical actor-critic architecture is composed of three elements: an actor, a critic, and the environment [111], as shown in Figure 4.3. The actor frequently observes the environment state and generates actions following a parameterized policy. The critic criticizes the actions selected by the actor based on a parameterized value function and the rewards fed back by the environment. The output of the critic is the estimated value of the environment state, which is then used to compute the temporal-difference (TD) error. Consequently, both the actor and the critic will update their functions (i.e., the policy function and the value function) using the TD error. The algorithm repeats this procedure until eiher convergence or completing the training iterations.

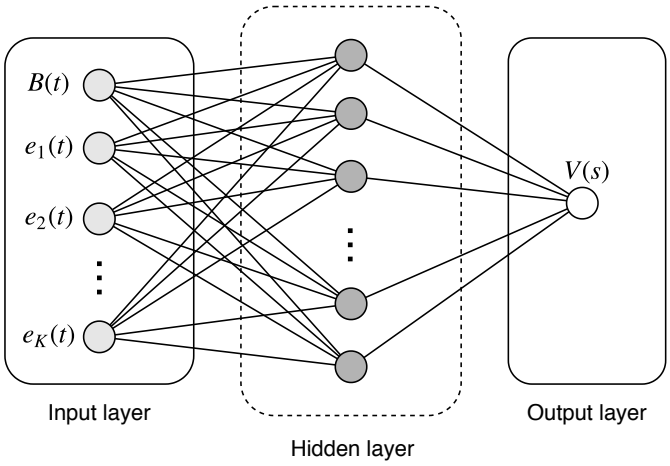

Figure 4.4: The structure of the value DNN with one hidden layer, which contains H_V neurons.

In particular, we adopt two sequential models of a DNN to approximate the policy and the value function in the actor and the critic, respectively. More specifically, policy π and value function $V^{\pi}(s)$ can be represented by $\pi_{\boldsymbol{\theta}}(s,a)$ and $V_{\boldsymbol{\omega}}(s)$ using DNNs with two different sets of parameters: $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$, respectively. These parameters are initialized randomly and then updated sequentially through the training process. The input of each network is the environment state. The policy DNN produces the probability distribution of all actions. On the other hand, the value-function DNN provides the estimated value of the environment state. In the following sections, we are going to use the phrases *system state* and *environment state* interchangeably.

The value-function DNN in the critic

This network consists of one input layer, one hidden layer, and one output layer, as shown in Figure 4.4. These layers are stacked into a sequential model. The input layer stores the system state in the form of a $1 \times (K+1)$ vector. The hidden layer is a regular densely-connected layer that contains H_V neurons and utilizes a rectified linear unit (ReLU) function for activation [115], as follows:

$$f(y) = \max(0, y) \tag{4.20}$$

where y is the estimated output of the layer before activation. Since the value-function network outputs the state value, the output layer contains only one neuron and uses a linear

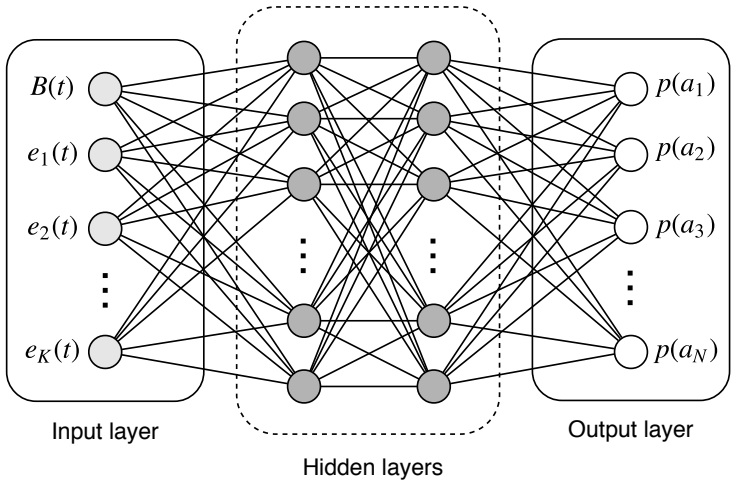

Figure 4.5: The structure of the policy DNN with two hidden layers, each of which contains H_P neurons. $N = |\mathcal{A}|$ and $p(a_n) = \Pr(a(t) = a_n | s(t)),\ n \in \{1, 2, \ldots, N\}$.

activation function to estimate the value of the system state. Network parameters $\boldsymbol{\omega}$ are optimized by stochastic gradient descent with the back-propagation algorithm to minimize the loss function. With this purpose, the loss function in the critic is the mean-squared error between the target value and the estimated value, calculated as

$$L(\boldsymbol{\omega}) = E[R(t) + \eta V_{\boldsymbol{\omega}}(s(t+1)) - V_{\boldsymbol{\omega}}(s(t))]^2 \tag{4.21}$$

where $\boldsymbol{\omega}$ denotes the value-function network's parameters. To minimize the loss function, parameters $\boldsymbol{\omega}$ can be updated in the direction of the gradient as

$$\Delta\boldsymbol{\omega} = \alpha_c \delta(t) \nabla_{\boldsymbol{\omega}} V_{\boldsymbol{\omega}}(s(t)) \tag{4.22}$$

where $\alpha_c > 0$ is the critic's learning rate; $\delta(t)$ denotes the TD error, which is given by

$$\delta(t) = R(t) + \eta V_{\boldsymbol{\omega}}(s(t+1)) - V_{\boldsymbol{\omega}}(s(t)) \tag{4.23}$$

The critic uses this TD error to guide the actor in generating actions to improve the network performance. For example, if an action produces a positive TD error, $\delta(t) > 0$, it will be preferred in the future when the system is in the same state, and vice versa.

The policy DNN in the actor

This network consists of one input layer, two hidden layers, and one output layer, as shown in Figure 4.5. The input layer has the same structure as the value-function

network. There are two hidden layers, each of which contains H_P neurons. Since the policy network outputs the probabilities of selecting actions for a given state, the output layer has the size of the action space. Moreover, the output layer uses a softmax activation function to produce the probability of each action in the action space, which can be defined as follows [1]:

$$g(y_a) = \frac{e^{y_a}}{\sum\limits_{a' \in \mathcal{A}} e^{y_{a'}}}, \ \forall a \in \mathcal{A} \tag{4.24}$$

where y_a is the estimated value of action a. Similar to the value-function network, the objective function in the actor is defined as

$$\begin{aligned} J(\boldsymbol{\theta}) = & \ E\left[V^{\pi}(s)\right] \\ = & \sum_{s \in \mathcal{S}} d^{\pi}(s) V^{\pi}(s) \\ = & \sum_{s \in \mathcal{S}} d^{\pi}(s) \sum_{a \in \mathcal{A}} \pi_{\boldsymbol{\theta}}(s,a) Q^{\pi}(s,a) \end{aligned} \tag{4.25}$$

where $d^{\pi}(s)$ denotes the state distribution for policy $\pi_{\boldsymbol{\theta}}$, and $Q^{\pi}(s,a)$ is the state-action value function, which is denoted as

$$Q^{\pi}(s,a) = E\left[\sum_{t=0}^{\infty} \eta^t R(t) | s(0) = s, a(0) = a, \pi\right] \tag{4.26}$$

Policy parameters $\boldsymbol{\theta}$ are optimized by gradient ascent with the back-propagation algorithm to maximize the objective function, as follows:

$$\Delta \boldsymbol{\theta} = \alpha_a \delta(t) \nabla_{\boldsymbol{\theta}} \ln \pi_{\boldsymbol{\theta}}(s,a) \tag{4.27}$$

where $\alpha_a > 0$ is the actor's learning rate. The learning rate of the actor is usually small in order to avoid oscillation in generating actions. It is worth noting that the TD error, $\delta(t)$, is provided by the value network in the critic process. Furthermore, the actor process uses an ϵ-greedy policy for selecting actions. Specifically, a random action, $a \in \mathcal{A}$, can be selected with probability $\epsilon \in [0,1]$, or an action is chosen based on the distribution output of the policy network with probability $1 - \epsilon$. With this strategy, the value of ϵ decays after every training iteration at decay rate d_ϵ. In both DNNs, the network parameters are randomly initialized using uniform Xavier initialization [116].

The training procedure of the proposed actor-critic deep learning approach is summarized in Algorithm 2.

Algorithm 2 The training procedure for the ACDL algorithm

Input: $\mathcal{S}$, $\mathcal{A}$, η, α_a, α_c, K, B_{max}, E_{max}, e, λ_e, λ_r, μ_s, μ_d, ϵ_{min}, ϵ_{max}, d_ϵ.

Output: $\pi^*(s) = \arg\max_{a \in \mathcal{A}} \{\pi_{\boldsymbol{\theta}}(s, a)\},\ \forall s \in \mathcal{S}$.

Initialize the two network parameters: $\boldsymbol{\theta}$, $\boldsymbol{\omega}$.
Initialize $\epsilon = \epsilon_{max}$.
for episode $ep = 1, 2, \ldots, M$ **do**
  Initialize the system state, $s(0)$.
  for step $t = 0, 1, \ldots, T-1$ **do**
    Set $\epsilon = \max(\epsilon \cdot d_\epsilon, \epsilon_{min})$
    Observe the current system state, $s(t)$, and estimate state value $V_{\boldsymbol{\omega}}(s(t))$.
    Choose action $a(t)$ according to ϵ-greedy policy $\pi_{\boldsymbol{\theta}}(s(t), a(t))$.
    Sample immediate reward $R(t)$.
    Observe the next system state, $s(t+1)$, and estimate state value $V_{\boldsymbol{\omega}}(s(t+1))$.
    if episode is terminated at $t+1$ **then**
      Set $\delta(t) = R(t) - V_{\boldsymbol{\omega}}(s(t))$
    else
      Set $\delta(t) = R(t) + \eta V_{\boldsymbol{\omega}}(s(t+1)) - V_{\boldsymbol{\omega}}(s(t))$
    end if
    Update the policy parameters: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$.
    Update the critic parameters: $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} + \Delta\boldsymbol{\omega}$.
    Update system state: $s(t) \leftarrow s(t+1)$.
  end for
end for

4.4 Numerical Results

In this section, we present the numerical simulations to validate the efficiency of the proposed ACDL scheme for joint user association and bandwidth allocation in dense networks with energy-harvesting base stations. We also compare the performance of our proposed method with that of other baseline schemes, e.g., an actor-only scheme [113], a critic-only scheme [114], a myopic policy [117], and a random policy. When using the myopic policy, the controller aims to maximize the instant reward received in the current time slot. With the random policy, the controller randomly selects an action in the action space based on the system state.

4.4.1 Simulation settings

The proposed ACDL algorithm was implemented using Python 3.6 with a TensorFlow deep learning library (Anaconda distribution, The Anaconda Inc., Austin, Texas, USA, 2018). We validated the performance of the proposed scheme under various simulation settings. We consider a mobile network with $K = 3$ green base stations powered by solar energy. The total system bandwidth, B_{max}, is composed of 10 channels, and each user can use one channel for its data service. Users are distributed randomly inside the network's service range, and the number of users is set at $U = 10$. We set the probability that a user issues a new data request to the network as $p_r = 0.8$. For DNN configuration, we use two sequential DNNs to model the policy function and the value function in the proposed algorithm. The architecture of each DNN is as described in Section 4.3. The number of neurons in the hidden layer of the value-function DNN is set at $H_V = 50$. We set the number of neurons in each hidden layer of the policy DNN at $H_P = 20$. For the training process, we use the Adam optimizer, an algorithm for first-order gradient-based optimization of a stochastic objective function [118], to iteratively update network weights after every training episode. Furthermore, the system state is initialized randomly, and the exploration rate, ϵ, is linearly reduced from 1 to 0.01. We use constant learning rates for the actor and the critic in our proposed method. We run the simulations several times to find the most appropriate learning rates, which can provide the best performance of the proposed algorithm. For comparison purpose, the neural-network structures of the DNNs in the actor-only and the critic-only methods, as well as the simulation parameters, are kept the same as those in our proposed scheme. We summarize basic simulation parameters for

Table 4.1: System model and algorithm parameters

Parameter	Description	Value
E_{max}	Battery capacity of a BS (packets)	20
λ_e	Mean harvested energy (packets/slot)	5
e	Required energy for transmitting data (packets)	3
μ_s	Mean channel holding time (slots)	2
μ_d	Mean cell sojourn time (slots)	1
α_a	Actor learning rate	0.0001
α_c	Critic learning rate	0.0005
γ	Discount factor	0.99
ϵ	Exploration rate	$1 \rightarrow 0.01$
d_ϵ	Decay rate	0.9999
M	Number of training episodes	200
T	Number of iterations per episode	2000

the system model and the proposed algorithm in Table 4.1.

4.4.2 Performance analysis

First, we investigated the convergence property of our proposed ACDL algorithm according to the number of training episodes when changing the number of steps per episode. Figure 4.6 illustrates the average rewards of the proposed scheme under different training iterations, T, while the number of episodes, M, increases gradually from 1 to 300. We obtained the final results by calculating the average rewards from 100 separate runs. As shown in the figure, fewer training steps provides the algorithm with a lower convergence speed, and thus, smaller rewards. Specifically, the RL agent needs more than 300 episodes to learn the optimal policy at $T = 1000$ steps per episode. When we increase T, the algorithm starts to converge faster, and the agent can learn the optimal policy in less than 200 episodes (e.g., with $T = 2000$). However, if each episode uses too many steps for training, the training process might take place over a very long time, and the algorithm might even converge to a locally optimum policy. Therefore, the maximum number of training episodes and the total number of training steps for each episode should not be too large or too small.

Then, we verified the robustness of our proposed scheme by comparing the per-

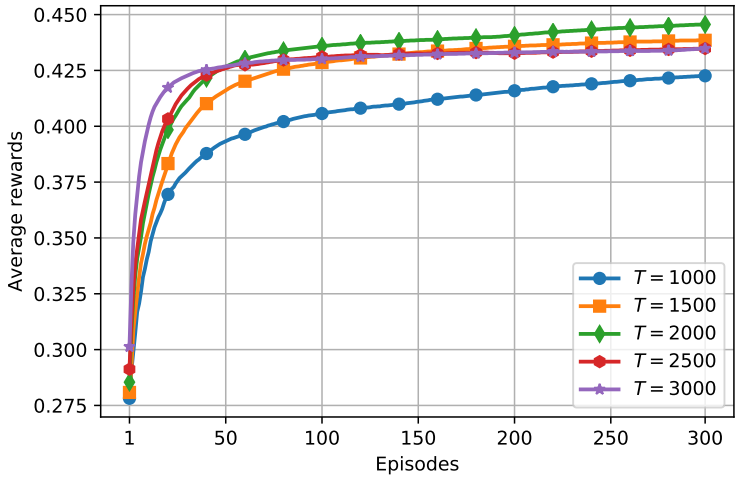

Figure 4.6: Convergence rate of the algorithm with different training steps in each episode.

formance of different learning and non-learning schemes. In this context, the total training episodes and the number of training steps per episode were fixed at $M = 200$ and $T = 2000$, respectively. We also performed 100 independent simulations for each learning scheme to get average rewards. Figure 4.7 shows the average estimates of the system rewards achieved by our proposed algorithm, compared with other deep learning algorithms, and the methods without a learning process. From the figure, we can see that the average reward from all the learning schemes rises quickly with an increase in the number of training episodes, and slows down gradually afterwards. The RL agents can learn the optimal policy after being trained with nearly 200 episodes. Among the three learning schemes, although the convergence speed of the actor-only method is the highest, needing fewer than 100 episodes until convergence, this scheme seems to converge to a locally optimum policy that leads to lower rewards. With the critic-only method, the agent selects an action based mostly on the Q-value of the system state, which might have a high variance [114], and thus, it might take more time to converge to the optimal solution. With the proposed ACDL algorithm, the agent can learn the optimal policy effectively by using the output of the critic network to guide the policy network in selecting actions, and thus, provides the best performance. Meanwhile, the rewards given by the two non-learning schemes remained unchanged because there is no learning process, and these schemes choose actions based mostly on the current state of the environment.

We further examined the effect of the total system bandwidth, B_{max}, on the per-

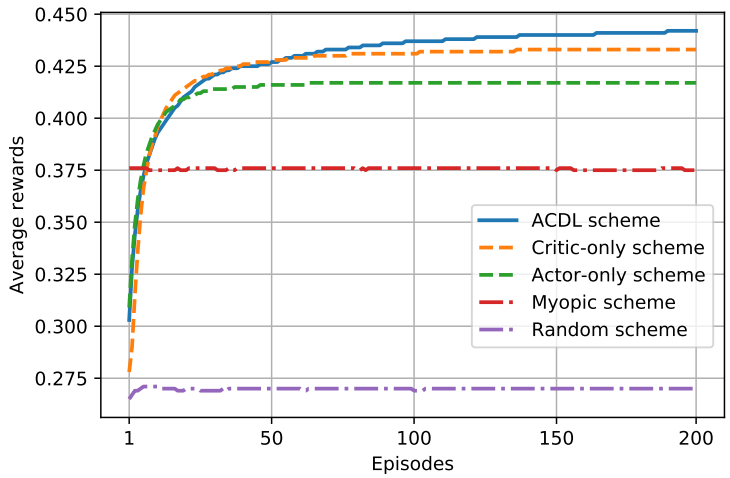

Figure 4.7: Convergence behavior of the different methods.

formance of the ACDL scheme, compared with other schemes, as depicted in Figure 4.8. In this scenario, the maximum number of channels that can be used for data services was set at $B_{max} \in \{2, 4, \ldots, 14\}$. As is shown in the figure, with more bandwidth, the network can provide data service to more users, and thus, receives better cumulative rewards. The system rewards increase quickly with an increase in the total system bandwidth when the number of channels is low (e.g., $B_{max} \leq 8$). When B_{max} is greater than 10, the increment in system bandwidth does not provide much better rewards. The reason might be the lack of energy at the base stations for data transmission using the allocated bandwidth. Furthermore, the proposed scheme outperforms other schemes in terms of average system rewards, since the ACDL algorithm can make efficient decisions on user association and bandwidth allocation while reserving system bandwidth for future use.

Similarly, Figure 4.9 demonstrates the effect of the harvested energy on the performance of the proposed scheme. The mean energy packets that a base station can harvest in each time step, λ_e, ranges from 1 to 5. As expected, with an increase in the number of energy packets that base stations can harvest in a time slot, the system rewards increase significantly. The reason is that if a base station can harvest more energy from solar power, it can preserve more renewable energy in the battery, which is then used to serve more users and get more rewards. In comparison to the non-learning schemes, the RL agents can learn the dynamic arrivals of harvested energy and data requests through interactions with the environment, and hence, can make efficient resource allocation decisions in the

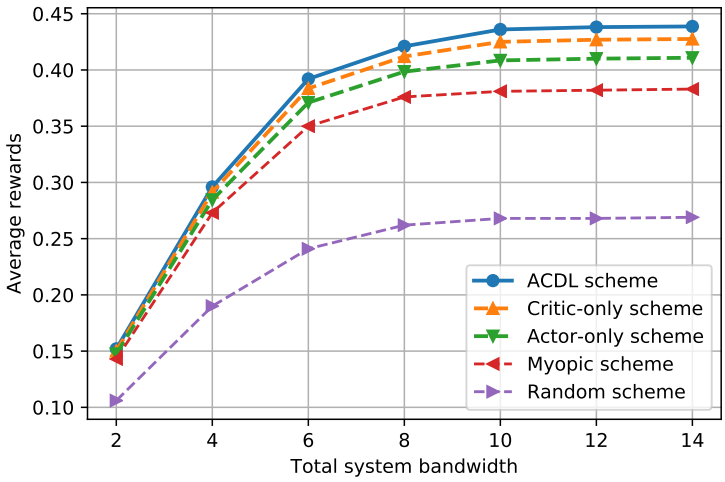

Figure 4.8: Average rewards according to total system bandwidth for the different methods.

current time slot. Consequently, more renewable energy can be saved for future use, which brings about greater system rewards. When using the myopic scheme, the controller tries to maximize the instant reward received in the current time step, which requires high energy consumption. This short-sighted action might cause the base stations to stay inactive in future time slots (from running out of stored energy). When using the random scheme, the controller randomly selects an action in the action space based solely on the system state. However, this kind of action selection might be inefficient due to variations in the arrival of renewable energy and data requests. For example, the controller might allocate too much bandwidth for data transmission in the current time slot, which causes the system to run out of bandwidth for use in future time slots.

Figure 4.10 shows the average system rewards according to the number of users. We set the number of users in the service range of the system at $U \in \{9, 10, \ldots, 17\}$. When the number of users in the system increases, the network will receive more data requests in every time slot, and thus, consumes more bandwidth and energy for data transmission. However, since the system bandwidth is restricted, and the battery capacity of each BS is finite, an increase in the number of users reduces the system rewards significantly. Hence, the controller needs to employ an actor-critic deep learning framework to obtain an efficient user-association and bandwidth-allocation policy based on the number of users. By using our proposed algorithm, the controller can follow the fluctuations in the request arrivals and in harvested energy, and thus, achieves greater cumulative rewards.

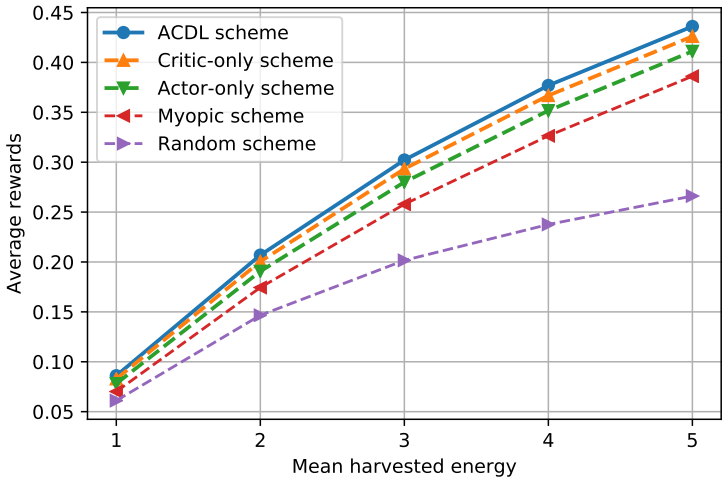

Figure 4.9: Average rewards according to the mean harvested energy for the different methods.

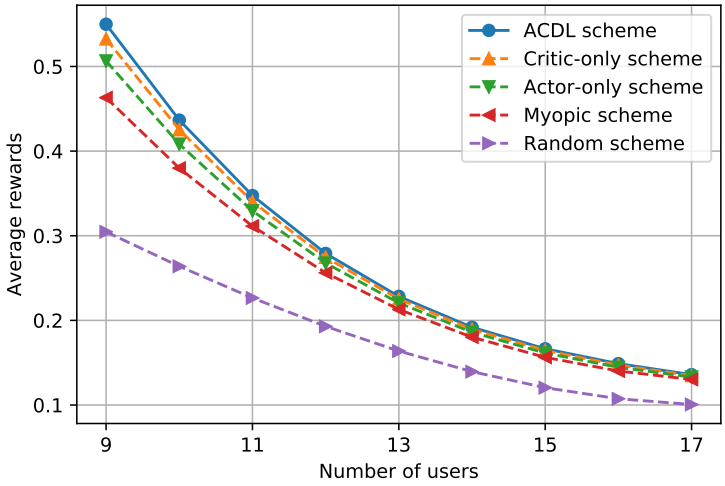

Figure 4.10: Average rewards according to the number of users for the different methods.

4.5 Conclusion

In this chapter, we investigated a deep learning framework for joint user association and bandwidth allocation in dense mobile networks with energy-harvesting base stations. More specifically, we formulated the optimization problem (adhering to constraints on harvested energy and bandwidth) as a Markov decision process. We then employed an actor-critic algorithm to find the optimal solution for maximizing the system rewards. We

further exploited deep neural networks to approximate the policy function and the value function, which allowed the algorithm to work with large state and action spaces. The agent of the ACDL algorithm can find the optimal policy through interactions with the environment. Consequently, the controller can effectively associate users with the base stations, and can then allocate bandwidth for their data transmissions based on the current state of the network. The simulation results show the advantage of the proposed solution in improving network performance in the long run. The bandwidth allocation problem in this work can be further extended to include the effects of the latency and packet losses in the returned rewards of the system, which might affect the overall performance of the network. However, it is essential to modify the DNN structures, the learning parameters, and the original state and action spaces to solve the more complicated problem.

Chapter 5

A Transfer Deep Q-learning Framework for Resource Competition in Virtual Mobile Networks with Energy-harvesting Base Stations

5.1 Introduction

The tremendous growth in data services for mobile communications (e.g., music, video, and games) has led to increasing demands for wireless resources in recent years. However, traditional network architectures might not satisfy these bandwidth- and time-intensive services owing to constrained wireless resources and increasing power consumption. As a consequence, both the academic and the industrial communities are focusing on developing efficient resource management schemes to improve spectrum utilization and energy conservation in the emerging mobile Internet [119]. In particular, network virtualization [9] and software-defined networking (SDN) [8] are considered the key technologies that are expected to enhance the network utility in terms of higher data rates, lower operational costs, and better resource utilization. Furthermore, energy-harvesting technology, which allows the harvesting device to obtain energy from ambient sources in the environment, is

considered a promising solution to energy conservation in dense cellular networks [120].

Wireless network virtualization (WNV) is a process of abstracting, slicing, isolating, and sharing radio resources in a virtualized way [121], which can promote better wireless-resource utilization and can reduce operational costs [122]. In mobile cellular networks, WNV allows virtual mobile operators (VMOs) to share the same network infrastructure (e.g., licensed spectrum, base stations) owned by a mobile network owner (MNO). On one hand, the VMOs can exploit the available infrastructure from the MNOs to provide customized services at competitive prices to their subscribers (i.e., mobile users). On the other hand, the MNOs can earn more revenue and attract more users by leasing their network resources to the VMOs. Consequently, the overall expenses of network deployment and operation can significantly decrease [9]. However, deploying WNV, in reality, is much more complicated due to the stochastic characteristics of wireless networks (e.g., time-varying wireless channels, signal attenuation, and user mobility).

One feasible solution is to separate the network control plane from the data plane by applying SDN to WNV, which can help to simplify the network management process, and thus, can improve the overall performance of the whole network [10, 123]. Zhang *et al.* [124] proposed a flexible architecture to establish a data delivery path for wireless network virtualization in an SDN-based environment, which can maximize the capacity of wireless virtualized networks with a QoS guarantee for data transmission. Meanwhile, the research on resource allocation in energy harvesting–based small-cell networks has also attracted increasing interest [125]. However, only a few studies consider the problem of resource leasing in wireless virtualized networks with energy harvesting.

In this paper, we study the resource leasing problem from the perspective of a VMO in an SDN-based virtualized mobile network that is powered by renewable energy. In this network, several VMOs lease radio resources from an MNO based on the service requests generated by their subscribers. Specifically, the MNO slices the spectrum resources into multiple sub-channels based on prior information about historical data usage of all VMOs, offering them for sale at different quality–price contract bundles. The VMOs, on the other hand, want to minimize their leasing costs while ensuring the best performance for their subscribers. Although there has been some excellent work on radio spectrum virtualization so far, there is little research considering reinforcement learning (RL)-based methods for WNV, especially for spectrum leasing and scheduling in energy harvesting–based small-cell networks. In particular, Chen *et al.* [126] considered the problem of resource

allocation in virtualized small-cell networks with full-duplex self-backhauls, which aim to maximize the total utility of all virtual network operators in terms of earned revenue and paid cost. Wu *et al.* [127] investigated a profit maximization problem for a cognitive-aided virtual network operator, in which the operator might choose to lease spectrum resources from the MNO, and access idle licensed bands at the same time. Fu and Kozat [128] proposed a stochastic game for WNV, in which the service providers (SPs) bid for the wireless resources via announcing their value functions. The resource-allocation game was decomposed into independent Markov decision processes, and the SPs update their value functions based on an online learning algorithm. Therefore, this paper aims to develop an autonomous resource-leasing scheme based on reinforcement learning, which is applicable in virtual mobile networks with energy-harvesting base stations. We model the problem as the framework of a Markov decision process, during which the VMOs compete for the radio resources needed to serve their users.

Among various RL methods, Q-learning has attracted a lot of attention in recent years thanks to its ability to solve many types of complicated decision-making problems with small-scale models [129–131]. However, it is necessary to integrate deep learning with Q-learning, referred to as deep Q-learning, or a deep Q-network [132], to deal with large state and action spaces. The agent in the deep Q-learning algorithm uses a deep neural network as a function approximator to estimate the Q-values of state–action pairs, which are then updated regularly from trial-and-error interactions with the environment [133]. In this paper, we implement a deep Q-learning algorithm that can find an optimal resource-leasing strategy for a VMO to maximize utility without prior information about system dynamics. To the best of our knowledge, using deep reinforcement learning in the design of a resource competition scheme in virtual mobile networks is a new research direction, and very little has been done in this direction so far. For example, Mijumbi *et al.* [134] proposed an autonomous resource allocation system based on artificial neural networks for virtual networks. In [135], the authors proposed an adaptive neuro-fuzzy system that uses both supervised and unsupervised learning for resource allocation in virtual networks. G. Sun *et al.* [121] formulated a resource slicing problem in virtualized radio access networks as a Markov decision process and solved this problem under a deep Q-learning framework. However, it could take time for classic, deep Q-learning algorithms to converge to the optimal solution, especially with large state and action spaces. We could deal with such an issue by combining deep Q-learning with transfer learning, in which historical data from

relevant problems might be used to speed up the learning process for a new problem [136]. Therefore, we propose a transfer deep–Q-learning (TDQL) framework for resource leasing in virtual networks, which utilizes a transferred deep Q-network to speed up the learning process.

In a nutshell, we propose a dynamic resource leasing scheme that can be applied to VMOs in virtual mobile networks so they can compete with each other for the radio resources to maximize utility. Our work focuses on maximizing the long-term utility of a VMO by employing the TDQL algorithm to train an agent to learn the optimal resource-leasing decisions from frequent interactions with the environment. The contributions of this paper are summarized as follows.

- We propose a novel, resource leasing and scheduling scheme that considers the dynamics of harvested energy, user demand, and resource pricing in a green virtual mobile network. In this network, the VMOs compete for radio resources, and try to guarantee long-term services to their subscribers within the constraints of harvested energy and resource sharing.

- We formulate the resource competition problem in this paper as the framework of a Markov decision process, during which the VMO determines the optimal leasing resources through interactions with the environment. More specifically, the VMO autonomously adjusts the resource requirements it announces to the MNO, which are based on the leasing price, the energy level at the base station, and the traffic demand from its subscribers. The goal is to maximize utility in the long run.

- We present a deep Q-learning algorithm, which is a combination of Q-learning and a deep neural network, to solve the formulated problem. Neural networks are used as function approximators to estimate the Q-value of each state–action pair. In the proposed scheme, we employ two well-known techniques (experience replay and fixed target network) to improve its stability.

- We further integrate the idea of transfer learning into the deep Q-learning–based resource-leasing strategy, which exploits learned knowledge from the past to enhance the convergence speed of the algorithm.

The rest of this paper is organized as follows. We first introduce the network model and the problem formulation in Section 5.2. In Section 5.3, we present the transfer

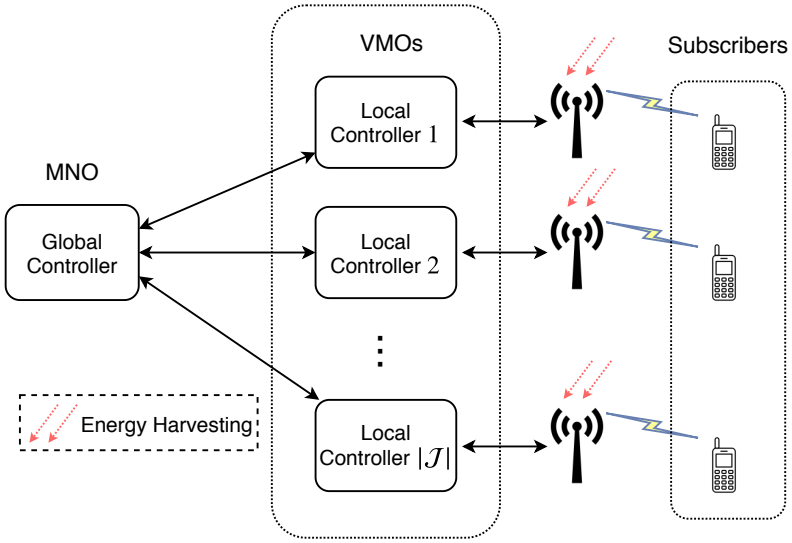

Figure 5.1: The considered wireless virtualized networks with energy-harvesting base stations in which the VMOs lease radio channels from the MNO to serve their subscribers. Each local controller represents one VMO.

deep Q-learning algorithm to solve the resource competition problem. Numerical simulation results are discussed in Section 5.4. Finally, Section 5.5 concludes this chapter.

5.2 The Network Model and Resource Competition Problem in Virtual Mobile Networks

5.2.1 The SDN–based Virtual Network Model

We consider a virtual mobile network (VMN) where several virtual mobile operators lease the infrastructure and spectrum resources from a single mobile network operator and offer customized services to their subscribers. In this paper, we focus on a radio resource–leasing scheme at the VMOs. We denote as $\mathcal{J}$ a set of VMOs that share N orthogonal channels (i.e., sub-channels [SCs]) owned by the MNO. We assume that each VMO is assigned a green base station (BS), which is powered solely by an energy harvester (e.g., by converting radio frequency energy and solar power into electrical energy). Each BS is equipped with a local controller that is connected to a global controller (GC) located at the MNO for network management, as shown in Figure 5.1. It is important to note that each local controller is supervised by a corresponding VMO. Furthermore, the aggregated

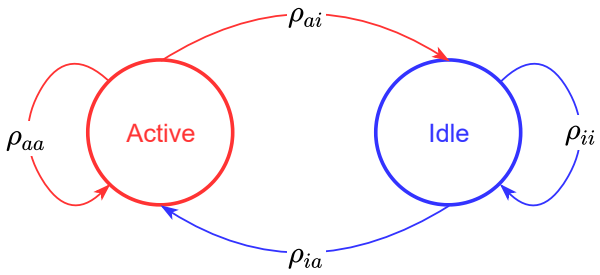

Figure 5.2: A two-state Markov model for user activity.

capacity of a BS (e.g., backhaul capacity, harvesting capability) is known in advance by both the MNO and the corresponding VMO that is assigned this BS. Since each VMO is assigned only one base station, the words BS and VMO can be used alternately in this paper.

We consider the downlink transmissions of such a network, where the VMOs use leased sub-channels to provide their customized data services (e.g., music streaming, video streaming, and gaming) to their users. The MNO first slices its radio resources into multiple virtual slices, and then dynamically assigns them to the VMOs based on an agreement between the MNO and each VMO. Moreover, the VMOs compete with each other for the wireless channels to maximize their utility. The network is assumed to operate on a time-slotted basis, in which each slot is denoted by t and is assumed to be of equal time duration (in seconds). During each slot, the environment statistics (e.g., harvested energy, service request arrivals, and spectrum prices) are assumed to remain unchanged. We assume that the action of base station assignment occurs only one time at the beginning of each contract period, which might contain a large number of time slots. We also assume in this paper that using more channels can provide the users with better quality for data services, yet would increase operational costs. We denote as $\mathcal{M}_j$ the set of mobile subscribers of VMO $j \in \mathcal{J}$, and m_j denotes a single user. In each time slot, a user might be in one of two possible states: active and idle. When a user is active, it randomly requests data services from the VMOs with different bandwidth requirements based on the type of subscribed service. We use a two-state discrete-time Markov chain to describe the state-switching process of each user, and the switching probabilities are ρ_{ai} and ρ_{ia}, where a and i denote active and idle, respectively, as shown in Figure 5.2.

The operation of the network is as follows. At the beginning of each time slot, the

MNO via the global controller initially reserves a minimum number of available channels for VMO j, $n_j^{[t]} \in (0, N)$, with a base price, p_0, for a radio channel based on a contract agreement between them. To guarantee isolation among the virtual resource slices that are allocated to the VMOs, the following constraint must hold:

$$0 < \sum_{j \in \mathcal{J}} n_j^{[t]} \leq N \tag{5.1}$$

The MNO also publishes the unit price for radio resources, $p_j^{[t]} \in [p_0, p_{max}]$, which might vary due to the demands of the VMO. The VMOs then announce to the MNO their required number of channels, and they need to consider the trade-off between user satisfaction and the price paid to the MNO. A VMO can benefit from an advance reservation base on a cheap price, which means that if the VMO requests fewer than $n_j^{[t]}$ channels at time t, the price is fixed at p_0; otherwise, the price will increase in proportional to the increased number of required channels. In reality, the initially reserved channels might not be sufficient for the requirements of the VMO due to high traffic. Another reason is that the MNO also wants to increase revenue by not assigning too many cheaply-priced channels to the VMOs. Therefore, each VMO might need to request extra channels according to its traffic density.

Since the radio resources are limited, VMO j then competes for wireless channels with other VMOs by announcing its resource requirements, $W_j^{[t]}\left(x_j^{[t]}\right)$, to the MNO based on users' requirements and the channel price announced by the MNO, where $x_j^{[t]}$ denotes the actual traffic demand at VMO j at time t. Afterward, based on the total amount of resources demanded by all the VMOs in the network, the GC allocates a finite number of radio channels to VMO j using a proportional fairness sharing scheme, as follows:

$$y_j^{[t]} = \min\left(W_j^{[t]}, \frac{N}{\sum_{j \in \mathcal{J}} W_j^{[t]}} W_j^{[t]}\right) \tag{5.2}$$

where $y_j^{[t]}$ is the number of SCs assigned to VMO j. This resource assignment scheme aims to achieve fairness among the VMOs. The VMOs in the network might have different strategies for competing with each other, and none has the information about the leasing schemes of the others.

5.2.2 Problem formulation

The entire process of resource leasing and scheduling in this paper takes place as follows.

(i) The MNO reserves a finite number of channels with a base price for each VMO based on the agreements between the MNO and the VMOs. The MNO also announces the unit resource price for the current time slot.

(ii) The VMOs announce their resource requirements to the MNO based on user demands, the leasing price, and the remaining energy of the base stations.

(iii) The MNO assigns radio channels to the VMOs using a fairness allocation mechanism.

(iv) The VMOs dynamically allocate the leased resources to their subscribers on a first-come-first-served basis. The base stations then use the allocated channels to transmit data to the end users.

The objective of the VMO is to maximize its user utility while minimizing the cost paid to the MNO.

In this paper, the users are divided into K groups of service classes that can be provided by any VMO in the network. The requirements of users may vary based on their subscribed services (e.g., gaming, video, or music). For each subscriber m_j of VMO j, let $x_{m_j}^{[t]}$ denote the number of sub-channels requested by the user in time slot t; hence, $\sum_{m_j \in \mathcal{M}_j} x_{m_j}^{[t]} = x_j^{[t]}$. Let c_k^{min} and c_k^{max}, respectively, denote the minimum and the maximum channel requirements of a user in class $k \in \{1, 2, \ldots, K\}$, so we have $c_k^{min} \leq x_{m_j}^{[t]} \leq c_k^{max}$. The role of VMO j is to compete for radio resources with other VMOs in order to provide its subscribers with the best performance. The channels assigned by the MNO are then allocated to the users according to the fairness allocation mechanism. Let $y_{m_j}^{[t]} \in \left[0, y_j^{[t]}\right]$ denote the number of sub-channels allocated to user m_j of VMO j, so we have $\sum_{m_j \in \mathcal{M}_j} y_{m_j}^{[t]} = y_j^{[t]}$. To ensure fairness among users, the VMOs first schedule the minimum channel requirements for each user, and then, they allocate the redundant channels (if available) to users as their requests arrive in order to improve service quality.

In this paper, we aim to develop an effective resource-leasing scheme at a VMO based on the information about the resource price, the request arrivals at the base station, and the energy level in the battery of the base station. Therefore, we define user and cost utilities to verify the effectiveness of the leasing strategy in terms of user satisfaction ratio and VMO surplus gain.

User Utility

We denote as $U_j^{[t]}$ the user utility for VMO j at time t. In this paper, the utility of subscriber m_j is represented by its level of satisfaction, which can be defined as the ratio of the number of allocated channels, $y_{m_j}^{[t]}$, to the number of requested channels, $x_{m_j}^{[t]}$, as follows:

$$U_{m_j}^{[t]} = \frac{y_{m_j}^{[t]}}{x_{m_j}^{[t]}} \tag{5.3}$$

where $U_{m_j}^{[t]} \in [0, 1]$. Therefore, the user utility for VMO j is the average utility of $|\mathcal{M}_j|$ subscribers, which is given by

$$U_j^{[t]} = \frac{1}{|\mathcal{M}_j|} \sum_{m_j \in \mathcal{M}_j} U_{m_j}^{[t]} \tag{5.4}$$

where $U_j^{[t]} \in [0, 1]$. Ideally, $U_j^{[t]} = 1$ indicates that the satisfaction ratio of the users at VMO j in time slot t is 100 percent.

Cost Utility

We denote as $C_j^{[t]}$ the cost utility for VMO j at time t. Given the unit price of sub-channels charged by the MNO, $p_j^{[t]}$, the normalized cost of purchasing resources from the MNO is given by

$$C_j^{[t]} = \frac{p_j^{[t]} y_j^{[t]}}{p_{max} N} \tag{5.5}$$

Consequently, the reward that VMO j can receive in a time slot is defined as the weighted sum of the average user utility and the cost utility, as follows:

$$R_j^{[t]} = \theta_u U_j^{[t]} - \theta_c C_j^{[t]} \tag{5.6}$$

where θ_u and θ_c are adjustable parameters that reflect the importance of user satisfaction and revenue, respectively, and $\theta_u + \theta_c = 1$. The VMO wants to minimize its leasing costs while providing users with the best performance. Hence, it needs to optimize the announcing of resources needed based on user demand, base station energy, and the price offered by the MNO.

In the scenario for this paper, each base station is equipped with an energy-harvesting device that can harvest renewable energy from ambient sources. We assume that the base station stores its harvested energy in a battery with a finite capacity, E_b, for

data transmissions. Furthermore, the energy packets collected by BS j during time slot t is denoted by $e_{h,j}^{[t]}$, which takes its value from a finite number of energy units, as follows:

$$e_{h,j}^{[t]} \in \{1, 2, \ldots, \vartheta\} \tag{5.7}$$

where $0 < \vartheta \leq E_b$. We assume that $e_{h,j}^{[t]}$ is a Poisson random variable with mean μ_e. The information about harvesting capability at a base station is available to the MNO and the VMOs in the network.

Due to the random characteristics of user requests, harvested energy, and leasing prices, the VMO needs to find the optimal resource announcement policy within the following constraints.

- *Energy constraint:* Let $e_{r,j}^{[t]} \in [0, E_b]$ denote the current energy level in the battery of BS j. The total energy consumption for data transmission must satisfy

$$e_{tr} W_j^{[t]} \leq e_{r,j}^{[t]} \tag{5.8}$$

  where e_{tr} denotes the energy consumption for providing services to the users when using one channel.

- *Resource constraint:* The amount of announced resource requirements should not exceed the requirements from the subscribers of the VMO:

$$W_j^{[t]} \leq x_j^{[t]} \tag{5.9}$$

  to ensure that the allocated resources can be fully utilized.

The problem of finding the optimal resource announcement policy of VMO j is given as follows:

$$\begin{aligned} \max_{W_j^{[t]}} \quad & \sum_{t=1}^{T} \gamma^{t-1} R_j^{[t]} \\ s.t. \quad & W_j^{[t]} \leq \min\left(\frac{e_{r,j}^{[t]}}{e_{tr}}, x_j^{[t]}\right) \end{aligned} \tag{5.10}$$

where $\gamma \in [0, 1]$ is a discount factor that reflects the present value of future rewards, and $\sum_{t=1}^{T} \gamma^{t-1} R_j^{[t]}$ is the accumulated sum of rewards on the time-horizon of length $T \in [1, \infty)$. The formulated problem can be solved by using value iteration-based dynamic programming methods if we have the information about the environment (i.e., the variations in harvested

Table 5.1: Notations

Symbol	Description
N	Number of shared channels
$\mathcal{J}$	The set of VMOs
$\mathcal{M}_j$	The set of subscribers for VMO $j \in \mathcal{J}$
K	Number of service classes
$[t]$	Index of a time slot (superscripted)
p_0	Base price of a radio channel
$p_j^{[t]}$	Unit price of a radio channel at VMO j at time t
$x_j^{[t]}$	Resource requirements for users of VMO j
$W_j^{[t]}$	Announcing resource as determined by VMO j
$y_j^{[t]}$	Number of channels allocated to VMO j
E_b	Battery capacity of each base station
$e_{r,j}^{[t]}$	Energy level in the battery of base station j
$e_{h,j}^{[t]}$	The amount of harvested energy at base station j
μ_e	Average harvested energy
e_{tr}	Energy consumption for transmitting data from a base station to a user using one channel
$R_j^{[t]}$	Immediate reward that VMO j receives at time t
γ	Discount factor

energy, resource prices, and request arrival rate), which is difficult to obtain in practice. Instead, we develop a learning-based algorithm that adopts an artificial neural network as a function approximator to solve the problem. With this algorithm, the agent can learn the optimal policy through interactions with the environment, as described in the next section. We provide the most used notations in Table 5.1 to make the paper more readable.

5.3 Deep Q-Learning for Resource Competition

Q-learning is a popular reinforcement learning algorithm where an agent tries to maximize its cumulative reward by regularly interacting with the environment through a decision-making process. The agent influences the environment by taking an action that causes the environment to transit from one state to another. The agent then receives a scalar signal as a reward for a *good* action (or a penalty for a *bad* action). Q-learning can

work effectively for problems with discrete state and action spaces. However, it is better to use deep Q-learning when dealing with high-dimensional state and action spaces. More specifically, a deep neural network, which is usually represented by a weight vector, $\boldsymbol{w}$, might be used as a function approximator to estimate the Q-value of any state–action pair. Therefore, in this section, we present a deep Q-learning algorithm for effective resource competition in wireless network virtualization.

5.3.1 Markov decision process

We reformulate the resource-competition problem in this paper as the framework of a Markov decision process (MDP), which is a generalized framework for modeling decision-making problems [137]. During this process, the agent can learn the optimal policy for resource announcement through a trial-and-error experience to maximize the accumulated sum of rewards. First, we define the state and action spaces of the MDP. The state of VMO j at time slot t is a combination of the remaining energy of the assigned BS, the unit price of the radio channels, and the current resource demands from subscribers, as follows:

$$s_j^{[t]} = \left(e_{r,j}^{[t]}, p_j^{[t]}, x_j^{[t]}\right) \tag{5.11}$$

where $e_{r,j}^{[t]}$ is the number of energy packets that are currently available at BS j for data transmissions, $p_j^{[t]}$ is the leasing price announced by the MNO, and $x_j^{[t]}$ is the total channel requirement at the beginning of the time slot. Based on the system state, the learning agent tries to select an action that can maximize its long-term reward. In our problem, the agent (i.e., the local controller) of VMO j has to decide on the number of channels it is going to request from the MNO in each time slot. Therefore, the action space of a VMO is denoted by

$$\mathcal{A} = \{a_0, a_1, \ldots, a_\zeta\} \tag{5.12}$$

where $0 \leq a_0 < a_1 < \ldots < a_\zeta \leq 1$. At the beginning of time step t, the agent observes the network state and takes action $a_j^{[t]} \in \mathcal{A}$, and then announces the total resource requirements, $W_j^{[t]} = a_j^{[t]} x_j^{[t]}$, to the MNO. Thereafter, the agent receives an instant reward, $R_j^{[t]}$, as defined by Equation (5.6). In particular, if $a^{[t]} = 0$, the VMO does not require resources from the MNO, and thus, no reward is achieved (i.e., $R_j^{[t]} = 0$). Otherwise, the MNO allocates a finite number of radio channels to the VMO, which are then used for data transmissions from the VMO to the end users. The action taken will also change the state of VMO j

from $s_j^{[t]}$ to $s_j^{[t+1]}$, which can be updated as follows. The energy level at base station j in the next time slot is given by

$$e_{r,j}^{[t+1]} = \min\left(e_{r,j}^{[t]} + e_{h,j}^{[t]} - e_{tr}y_j^{[t]}, E_b\right) \tag{5.13}$$

In this work, the MNO sets the resource price according to the demand of the VMOs. We assume that the leasing price of each SC in the next time slot is determined by a pricing function, which is defined by the MNO as follows:

$$p_j^{[t+1]} = p_0 + \frac{\beta_j^{[t]}}{N}\left(\sum_{j\in\mathcal{J}} y_j^{[t]}\right)^{\tau} \tag{5.14}$$

where p_0 is the base price of a radio channel, $\tau \geq 1$ is a constant, and $\beta_j^{[t]}$ is a non-negative coefficient to implement elastic pricing (e.g., the MNO tends to set a higher price for higher demand and lower network capacity). In this paper, the price coefficient, $\beta_j^{[t]}$, is defined as

$$\beta_j^{[t]} = \beta\left[y_j^{[t]} - n_j^{[t]}\right]^+ \tag{5.15}$$

where β is a positive constant, and $[\cdot]^+ = \max(0, \cdot)$. Meanwhile, the channel requirements in the next time slot are dependent on the distribution of user activities. According to the VMO's state and the received reward, the agent can choose better actions in future slots, which might provide the agent with better rewards.

This paper aims to find the optimal resource announcement policy to maximize the total discounted reward from the current step. To estimate the long-term reward of the VMO, we employ a state–action value function (i.e., the Q-value function), $Q(s, a)$, which is defined as the expected sum of rewards when it is in state s and takes action a, as follows:

$$Q(s,a) = \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} R_j^{[t]} | s_j^{[1]} = s, a_j^{[1]} = a\right] \tag{5.16}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator. Consequently, our objective is to find the optimal action, a^*, in the current time slot to maximize the Q-value function, as follows:

$$a^* = \arg\max_{a\in\mathcal{A}}\{Q(s,a)\} \tag{5.17}$$

The optimal action can be found by using the Q-learning algorithm, through which the state–action value function can be updated in each time slot with learning rate $\alpha \in (0, 1)$, as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[R + \gamma \max_{a'\in\mathcal{A}} Q(s',a') - Q(s,a)\right] \tag{5.18}$$

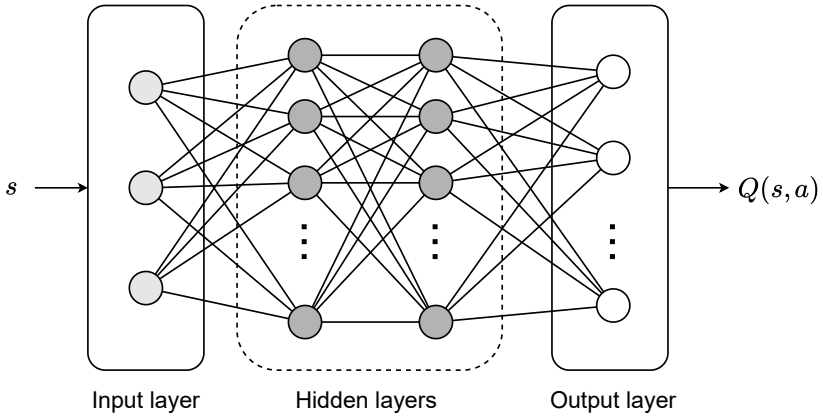

Figure 5.3: The architecture of the proposed Q-network in this paper.

where s' and a' are the next state and action, respectively. R is the instant reward that the VMO receives by serving the users in a time slot. With proper configuration, the Q-value function can converge to the optimal Q-value, from which the agent can select the optimal action to influence the environment in each slot. However, the original Q-learning algorithm might have wide variances in function approximation, which can cause the algorithm to converge to the locally optimal policy, especially when the size of the problem increases [138]. Therefore, we go further and use a neural network with weight $\boldsymbol{w}$ to approximate the Q-value function, denoted by $Q(s, a; \boldsymbol{w})$, so the proposed solution can work efficiently with large state and action spaces.

5.3.2 Deep Q-network Training

In this section, we describe the architecture of the deep neural network in the proposed deep Q-learning algorithm. We employ a feed-forward neural network (FNN) to approximate the Q-value function, which is thus named a Q-network. This neural network contains one input layer, several hidden layers, and one output layer. The FNN uses the VMO's state as the input to produce the Q-value of any state–action pair at its output. The sequential model of the proposed network is illustrated in Figure 5.3. Since the input layer is used to store the state of a VMO, it consists of three neuron units that represent the three elements in each state. The hidden layers are fully connected layers that contain finite neurons, which use a rectified linear unit (ReLU) function as a non-linear activation

function. Hence, the output vector of the hidden layers is given by

$$\boldsymbol{z} = \max(\boldsymbol{w}.\boldsymbol{s} + \boldsymbol{b}, 0) \tag{5.19}$$

where $\boldsymbol{w}$ is the weight vector of the FNN, and $\boldsymbol{b}$ is a bias. The output layer of the FNN matches the output values of the hidden layers to the estimated Q-value of each state–action pair (given the state) by using the linear activation function, and hence, the size of this layer is the size of the action space, which is $\zeta + 1$.

By training the FNN, the network parameters are iteratively optimized to minimize the loss function, which is defined as the mean square error between the target value and the current Q-value, as follows:

$$L(\boldsymbol{w}) = \mathbb{E}\left[\left(R + \gamma \max_{a'} Q(s', a'; \boldsymbol{w}) - Q(s, a; \boldsymbol{w})\right)^2\right] \tag{5.20}$$

where $R + \gamma \max_{a'} Q(s', a'; \boldsymbol{w})$ is the target value. Furthermore, to alleviate the effect of data correlations and non-stationary targets on the stability of the learning process [139], we also apply two well-known techniques in our work, namely, *experience replay* [140] and *fixed target network* [138]. With the fixed-target-network technique, we use another neural network with network weight $\boldsymbol{w}^-$ to compute the target value, and the network parameters are kept unchanged during a finite number of training iterations. With the experience-replay technique, the transitions (s, a, R, s') are stored in a *replay buffer*, $\mathcal{D}$, from which random mini batches are selected to train the Q-network, instead of using consecutive samples, as follows:

$$L(\boldsymbol{w}) = \mathbb{E}_{\mathcal{D}}\left[\left(R + \gamma \max_{a'} Q\left(s', a'; \boldsymbol{w}^-\right) - Q(s, a; \boldsymbol{w})\right)^2\right] \tag{5.21}$$

The target network and the Q-network have the same structure, and the target network's parameters are frequently replaced by those of the Q-network during the training process, $\boldsymbol{w}^- \leftarrow \boldsymbol{w}$. The weight vector $\boldsymbol{w}$ is updated by using stochastic gradient descent to minimize the loss function in the direction of the gradient, as follows:

$$\Delta \boldsymbol{w} = \alpha \delta \nabla_{\boldsymbol{w}} Q(s, a; \boldsymbol{w}) \tag{5.22}$$

where δ denotes the temporal different (TD) error between the target value and the current Q-value, which is given by

$$\delta = R + \gamma \max_{a'} Q\left(s', a'; \boldsymbol{w}^-\right) - Q(s, a; \boldsymbol{w}) \tag{5.23}$$

The agent uses the TD error to adjust the network parameters in the direction that improves system performance. Moreover, the agent selects actions according to an ϵ-greedy policy, where $\epsilon \in [0, 1]$ is the *exploration rate* [141]. With this policy, exploration rate ϵ decays in each iteration of the training process at rate d_ϵ. The algorithm repeats the Markov decision process until convergence.

Furthermore, we exploit the idea of transfer learning to increase the learning speed of the agent in our problem by making use of historical learning data. Instead of learning from scratch, the local controller might directly choose proper actions at the very beginning, based on the learned strategy. In deep Q-learning, the transferred knowledge could be the weights of a well-trained Q-network or the Q-values of state–action pairs. In this paper, we utilize a Q-network that is well trained in historical moments or in a relevant environment to help the agent choose better actions at the initial stage of the learning process. For example, the learning agent can use a Q-network that has been trained in a system with the same state space, action space, and reward function as in the target system. It is also possible to use the existing Q-network of the current system, which has been learned in historical periods, to train the current network. By transferring the learned knowledge, the deep Q-learning algorithm could exploit the relevancy in the harvested energy model and service request model to speed up the continuous learning process of the agent in the new environment. Specifically, the overall Q-value of each state-action pair at step t, given state $s^{[t]} = s$, is computed as

$$Q_o(s, a) = \xi Q_{tf}(s, a) + (1 - \xi)\, Q(s, a)\ \forall a \in \mathcal{A} \tag{5.24}$$

where Q_{tf} and Q are the transferred Q-network and the new Q-network, respectively, and $\xi \in (0, 1)$ is a transfer rate that determines the contribution of the transferred Q-network to the overall Q-value. The impact of the transferred Q-network on the performance of the new Q-network decreases over time with decay factor d_ξ. According to the ϵ-greedy policy, the agent might select a random action with probability ϵ. Otherwise, the action is selected based on the overall Q-values of all state–action pairs, given the state, as follows:

$$a^{[t]} = \arg\max_{a \in \mathcal{A}} \left\{ Q_o \left(s^{[t]}, a \right) \right\} \tag{5.25}$$

The new Q-network, $Q(s, a; \boldsymbol{w})$, still frequently updates weights based on the deep Q-learning algorithm.

The training procedure in the proposed transfer deep Q-learning algorithm for resource competition in virtual mobile networks is described in Algorithm 3.

5.4 Performance Analysis

5.4.1 Simulation Settings

In this section, we present numerical simulations to assess the performance of our proposed resource-leasing approach under various configurations. We performed the simulations by using Python-integrated software with TensorFlow deep-learning libraries (Python 3.7, Anaconda 2019 distribution, The Anaconda Inc., Austin, Texas, USA, 2019). Here, we simulated a virtual mobile network consisting of J VMOs that lease $N = 15$ orthogonal channels from an MNO, where $J \in \{2, 3\}$. Each VMO provided $K = 3$ types of data services (i.e., music, videos, gaming) to $|\mathcal{M}_j| = 5$ subscribed users. We assume that a subscriber requests data for only one type of service at each time step, and that an active user randomly requests data for any of the three service classes with the same probability. The minimum channel requirement for all services was 1, and the maximum channel requirement was from the set $\{1, 2, 3\}$ corresponding to the three service types. It is worth noting that the users always want to receive the best service quality, and thus, they would request the maximum number of resources for their services. The state of each user in a time slot followed a discrete-time Markov process with transition probabilities $\rho_{ai} = \rho_{ia} = 0.2$.

Furthermore, we assumed that the MNO decides the minimum resources reserved for VMO j at the beginning of each time slot based on historical user distributions from its subscribers, as follows:

$$n_j^{[t]} = \left\lceil \frac{\rho_{ia}}{\rho_{ia} + \rho_{ai}} \times C_{|\mathcal{M}_j|} \right\rceil \tag{5.26}$$

where $\lceil \cdot \rceil$ denotes the ceiling function, $\frac{\rho_{ia}}{\rho_{ia}+\rho_{ai}}$ is the probability that a user is active in a time slot, and $C_{|\mathcal{M}_j|}$ denotes the minimum channel requirements at the VMO when all the users are active. For the pricing function, we assumed that the base price of each channel is 1 pricing unit, and we set the pricing parameters to $\beta = 0.3$ and $\tau = 1.5$. The unit price of radio channels ranged from 1 to 5 pricing units. Regarding energy harvesting, we set the average harvested energy in each time slot at a base station at $\mu_e = 5$ energy packets, and the energy storage at a base station has a capacity of $E_b = 20$ energy packets. We set the

Algorithm 3 TDQL — Training procedure

Input: $\mathcal{M}$, K, E_b, e_{tr}, μ_e, θ_u, θ_c, $\mathcal{S}$, $\mathcal{A}$, N, γ, α, T, ϵ_{init}, ϵ_{min}, d_ϵ, Q_{tf}, ξ_{init}, d_ξ

Output: Q-network parameter $\boldsymbol{w}$

Initialize $\boldsymbol{w}$ randomly and set $\boldsymbol{w}^- = \boldsymbol{w}$
Initialize exploration rate $\epsilon = \epsilon_{init}$
Initialize transfer rate $\xi = \xi_{init}$
Initialize replay memory $\mathcal{D}$
repeat
  Select initial state $s \in \mathcal{S}$
  for each step $t \in [1, 2, \ldots, T]$ **do**
    Set $\epsilon = \max(\epsilon \times d_\epsilon, \epsilon_{min})$
    Observe current state s
    Compute overall Q-value $Q_o(s, b)$ $\forall b \in \mathcal{A}$
    Execute action a based on the ϵ-greedy policy
    Obtain immediate reward R
    Observe next state s'
    Store the transition $\langle s, a, R, s' \rangle$ in memory $\mathcal{D}$
    Take random mini batches $\langle s_i, a_i, R_i, s_{i+1} \rangle$ from $\mathcal{D}$
    for i in length of mini batches **do**
      Estimate current value $Q(s_i, a_i; \boldsymbol{w})$
      Calculate $R_i + \gamma \max_{a'} Q\left(s_{i+1}, a'; \boldsymbol{w}^-\right)$
    end for
    Update Q-network parameter $\boldsymbol{w}$
    if $t \leq T - 1$ **then** update $s \leftarrow s'$
  end for
  Update target network parameter $\boldsymbol{w}^- \leftarrow \boldsymbol{w}$
  Update $\xi \leftarrow \xi \times d_\xi$
until convergence

number of energy packets required for transmitting data using one channel at $e_{tr} = 2$.

As for the proposed TDQL algorithm, action space $\mathcal{A}$ was quantized into 11 levels between 0 and 1 in the simulation. For the state space, each element of a state is an integer located between its minimum and maximum values: $e_{r,j}^{[t]} \in [0, E_b]$, $p_j^{[t]} \in [p_0, p_{max}]$, and $x_j^{[t]} \in [0, N]$. We used constant learning rate $\alpha = 0.01$, and the discount factor was set to $\gamma = 0.99$. The weighting parameters of the reward function introduced in (5.6) were set to $\theta_u = 0.8$ and $\theta_c = 0.2$, which indicate that the VMOs set a higher priority on the service satisfaction of their subscribed users. The Q-network or the target network contains one hidden layer of 100 neurons. For training deep neural networks, we used an adaptive optimization algorithm (i.e., the Adam optimizer [118]) to frequently update the weights of the Q-network after each episode of the training process. We set the size of replay memory $\mathcal{D}$ and the size of each mini batch to 2000 and 100, respectively. With the ϵ-greedy policy, the initial exploration rate and its decay rate were set to $\epsilon_{init} = 1$ and $d_\epsilon = 0.9999$, and the minimum exploration rate was $\epsilon_{min} = 0.01$. Also, the transferred Q-values were computed with the initial transfer rate, $\xi_{init} = 0.5$, which decayed at rate $d_\xi = 0.99$. In our simulation, we first trained a Q-network from scratch, and then used this network as a transferred model to train another Q-network with different environment setups (e.g., changing the average harvested energy, the total amount of spectrum resources, and the number of subscribers). We trained the network over 200 episodes, each of which contained $T = 2000$ time slots. For comparison purposes, we set the same initial states (i.e., the same energy level, the same resource price, and the same channel requirements) for all the VMOs. Here, we obtained the final results by averaging a large number of independent runs.

5.4.2 Results and Discussion

First, we examine the convergence property of the proposed TDQL algorithm during the training process. We specify the convergence condition of the TDQL algorithm based on the convergence of the average rewards. In each time slot, a VMO can receive an immediate reward, $R_j^{[t]}$. We regularly computed the average value of the rewards that the VMO received in T slots of every episode, denoted as R_{ep}, as follows:

$$R_{ep} = \overline{R_j^{[t]}} = \frac{1}{T} \sum_{t=1}^{T} R_j^{[t]} \tag{5.27}$$

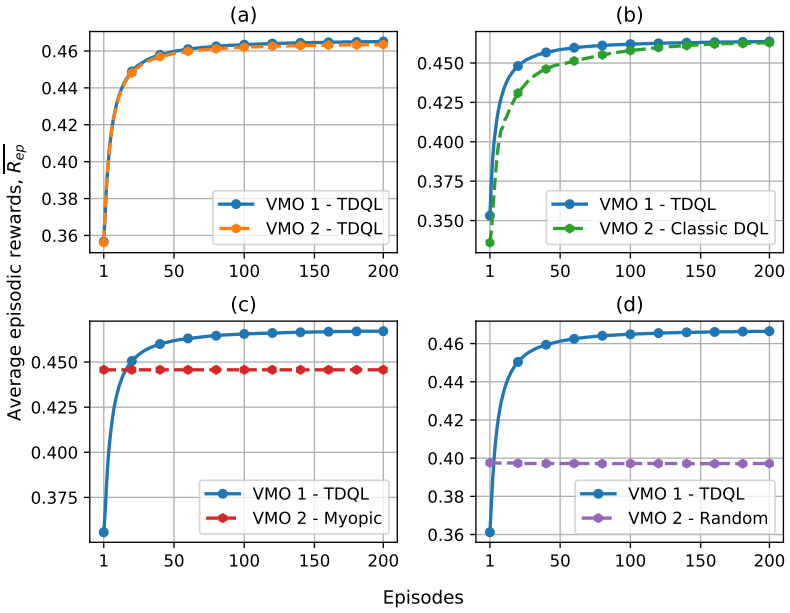

Figure 5.4: Convergence behavior of the proposed scheme in a wireless virtualized network consisting of two VMOs.

We further calculated the average episode reward at episode e, $\overline{R_{ep}}(e)$, as follows:

$$\overline{R_{ep}}(e) = \frac{1}{e}[R_{ep}(1) + R_{ep}(2) + \cdots + R_{ep}(e)] \tag{5.28}$$

With this method, we aimed to make the results look flat (i.e., little or no fluctuation). In this paper, the convergence condition is defined as $|\overline{R_{ep}}(e) - \overline{R_{ep}}(e-1)| < 0.0001$. Therefore, the TDQL algorithm keeps training the Q-network until it meets the convergence condition or reaches the maximum number of training episodes.

Figure 5.4 illustrates the convergence process of the proposed algorithm in terms of average episode reward for each VMO in a network that has two VMOs. In this system, the first VMO only uses the TDQL algorithm for resource leasing. Meanwhile, the second VMO might use different resource-leasing strategies, such as learning and non-learning methods. Different policies might have different effects on the average episode reward. For example, the second VMO might use the proposed TDQL algorithm or a classic deep Q-learning (DQL) algorithm for spectrum leasing, as is shown in Figures (5.4a) and (5.4b), respectively. From the figures, we observe that the rewards of the learning methods increase quickly in the first 50 episodes, and then gradually converge to the optimal value. If the second VMO also uses the TDQL algorithm to train its agent, its learning process has

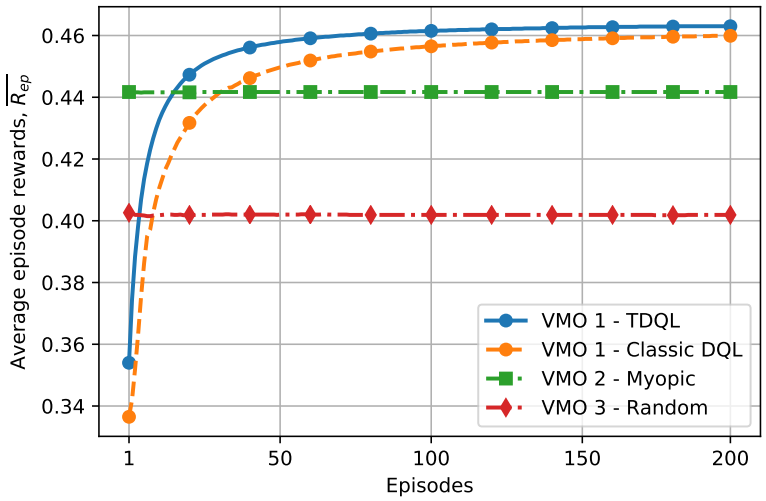

Figure 5.5: Convergence behavior of the proposed algorithm in a wireless virtualized network with three VMOs.

the same convergence speed as that of the first VMO, and the agent can find the optimal resource-leasing policy after being trained for more than 100 episodes. The rewards for the VMO that uses the classic DQL method are lower than those of the VMO using the TDQL solution. The agent also needs more episodes to learn the optimal policy. The reason is that the agent in the TDQL algorithm can learn faster by exploiting a well-trained network in selecting actions to influence the environment. On the other hand, the classic DQL method trains its agent from scratch, and hence, it needs to experience more trials as well as errors to learn. VMO 2 can also use non-learning strategies, such as the myopic and random schemes, as shown in Figures (5.4c) and (5.4d), respectively. With a myopic policy, the VMO wants to maximize the immediate reward that it receives in the current time slot. This method is equivalent to the proposed scheme when the discount factor is set to zero ($\gamma = 0$). When using the random policy, the VMO chooses an action randomly based on its current state. Therefore, the results given with the two non-learning schemes remain unchanged when the number of training episode increases.

Similarly, Figure 5.5 compares the performance of the proposed method with the classic DQL in a virtualized network consisting of three VMOs. In this scenario, the second and third VMOs rent resources from the MNO by using myopic and random schemes, whereas the first VMO exploits the TDQL algorithm or the conventional DQL algorithm in competing for resources against the other two. As depicted in the figure, the average

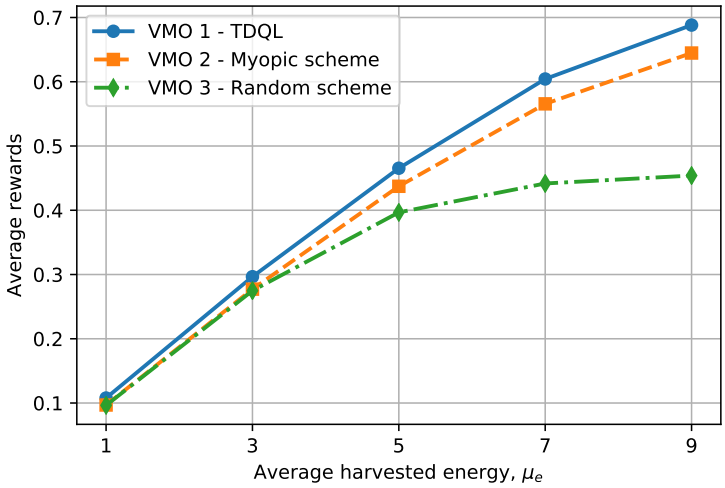

Figure 5.6: Average rewards for the VMOs in the network based on average harvested energy.

episode reward for the first VMO increases with the increase in the number of training episodes. Furthermore, the TDQL algorithm uses the pre-trained Q-network from a similar environment to guide its agent in choosing actions during the training process. Hence, it converges much faster than the conventional DQL method. As a result, the TDQL agent can learn the optimal policy efficiently and provides the VMO with the best performance. For the sake of simplicity, in the following simulations, we mainly discuss the results obtained in the scenario with three VMOs.

We further inspected the impact of harvested energy on the performance of the resource leasing schemes by varying the average harvested energy at the base station from 1 to 9, as shown in Figure 5.6. For each value of μ_e, we first trained the Q-network with 150 episodes, and we then tested the performance of the system by averaging the rewards over $10,000$ time slots. As observed from the figure, the average reward achieved by the VMOs increase significantly with an increase in the number of energy packets that the base stations can harvest in a time slot. Obviously, with more energy to be preserved in the battery, the base station (i.e., the corresponding VMO) can provide its customized services to more users, which leads to better results. Furthermore, the agent of the learning approach can predict the arrival of harvested energy as well as the resource demands and channel price based on the state of the VMO. Therefore, it can select appropriate actions to influence the environment effectively and receives more rewards. As a result, the proposed scheme

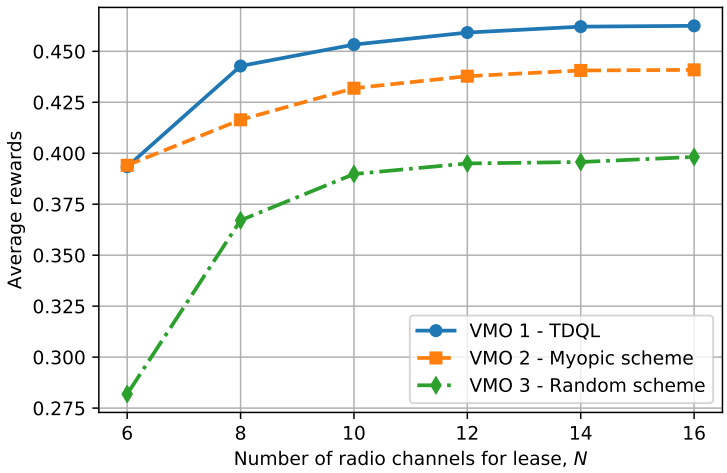

Figure 5.7: Average rewards for the VMOs based on the number of radio channels for lease.

provides the VMO with the best performance compared with other solutions. Meanwhile, the myopic-based VMO tries to serve its users with the best performance in the current time slot by asking the MNO for as many channels as possible. As a result, it would use a lot of energy for data transmission in the current step, and does not have enough energy for future use. Hence, the rewards are lower than with the proposed approach.

Figure 5.7 presents the rewards from an increase in the number of total radio channels that are available for lease, N. In this case, N ranged from 6 to 16 channels, and the average harvested energy was set at $\mu_e = 5$ packets. The returned rewards of the VMOs are low when there are fewer channels for lease from the MNO. The reason is that if the channel capacity of the system is low (i.e., N has small value), the VMOs are not allocated enough resources to serve their subscribed users, and hence, obtain fewer rewards. When $N < 12$, the rewards for all VMOs grow significantly with an increase in the number of available channels for lease. When $N > 12$, they are still rising but at a much lower rate. That could be because the base stations do not have enough energy, so the VMOs cannot request more resources from the MNO. Besides, the proposed TDQL approach outperforms the other methods, since the agent of the TDQL-based VMO selected better actions based on its learning experience, which utilizes the limited resources more efficiently.

We further examined the impact of traffic demand on the performance of the proposed TDQL approach when the number of subscribed users was set at $M \in \{3, 4, 5, 6, 7, 8\}$, as shown in Figure 5.8. From the figure, we can see that growth in the number of subscribers

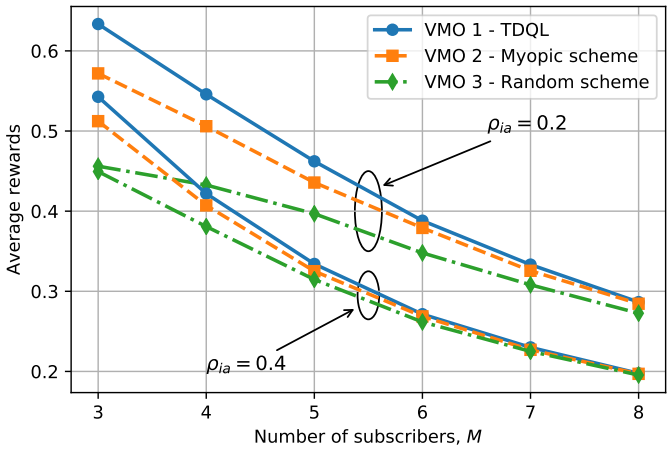

Figure 5.8: Average rewards for the VMOs based on the number of users and different values of ρ_{ia}, when $\rho_{ai} = 0.2$.

to each VMO causes the rewards to reduce significantly. Similarly, in a time slot where the probability that a user transits from the idle state to the active state rises from $\rho_{ia} = 0.2$ to $\rho_{ia} = 0.4$, the VMOs might need more channels to satisfy user demand. However, since the total number of radio channels in the system, and the energy storage of the base stations, are restricted, the VMOs' rewards also decrease. Again, the TDQL agent can learn the dynamics in the arrivals of harvested energy and user activity, and thus, provides VMO 1 with the highest performance. Meanwhile, the VMO that uses the myopic strategy always requests a large amount of resources to maximize the instant reward in the current time slot, which might cause its assigned base station to stay inactive in the future due to a lack of energy. Consequently, the results of the two non-learning VMOs are not as good as that of VMO 1.

In Figure 5.9, we present the impact of pricing parameter τ on the rewards of the VMOs when the weight of the user utility, θ_u, was set at $\theta_u = 0.8$ or $\theta_u = 0.5$. As observed from the figure, the increment of τ in the pricing function causes the VMOs to pay more to the MNO, and that reduces the average reward for the whole system. Furthermore, if we reduce the weight of the user utility in the reward function (i.e., the value of θ_u changes from 0.8 to 0.5), the impact of τ on the rewards also increases. While the rewards for all the VMOs decrease, the performance of the TDQL algorithm is still better than the other approaches, since the TDQL agent automatically adapts to the changes in the environment.

Figure 5.10 shows the convergence of the proposed algorithm in terms of leasing

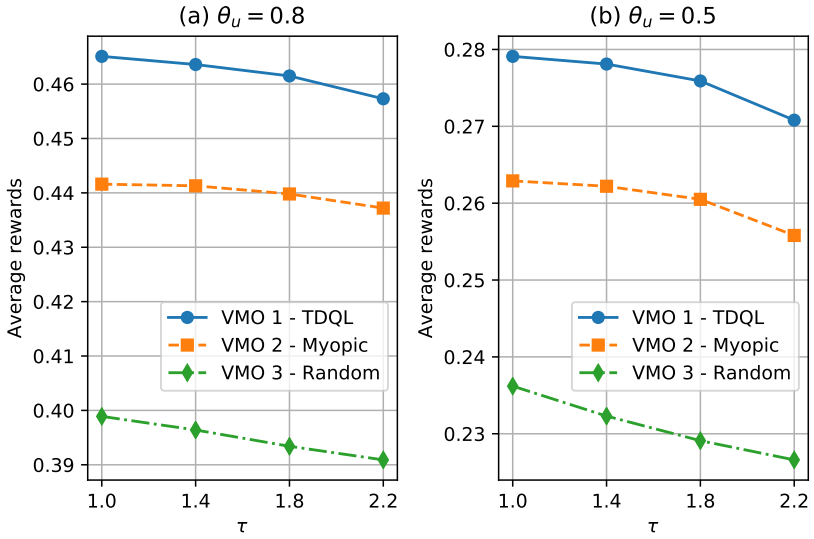

Figure 5.9: Average rewards for the VMOs based on pricing parameter τ with different values of θ_u.

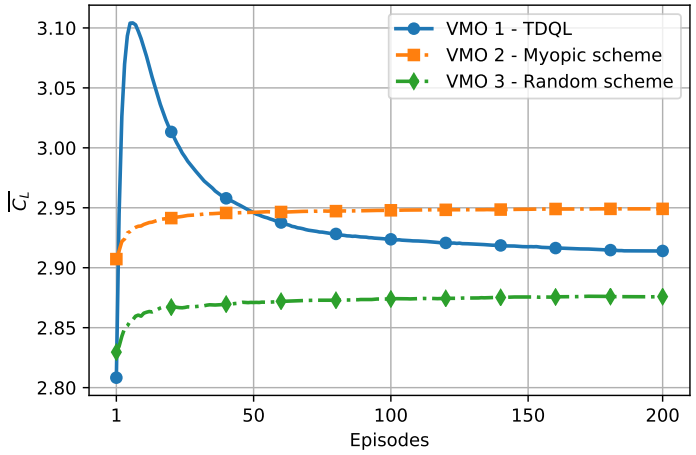

Figure 5.10: Convergence of the algorithm in terms of leasing cost paid by the VMOs.

cost paid by the VMOs. In this simulation, we also consider a system consisting of three VMOs, each of which uses a specific leasing strategy. The average leasing cost that a VMO has to pay the MNO in episode e is denoted by $\overline{C_L}(e)$ and is computed just like the average episode reward in Eqs. (5.27) and (5.28), as follows:

$$\overline{C_L}(e) = \frac{1}{e}[C_L(1) + C_L(2) + \cdots + C_L(e)] \tag{5.29}$$

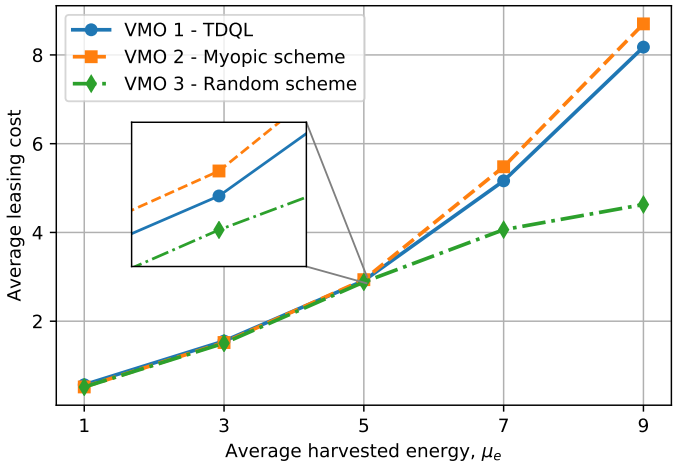

Figure 5.11: Average leasing cost of the VMOs based on average harvested energy.

where C_L is an episodic cost that the VMOs pay in each episode:

$$C_L = \frac{1}{T}\sum_{t=1}^{T} p^{[t]} y^{[t]} \tag{5.30}$$

For the sake of simplicity, we removed the index notation j in the above equations, since they can be used to compute the cost for any VMO in the system. As can be seen from the figure, the leasing costs for all VMOs grow quickly in the first 20 episodes. For VMO 1, its total leasing cost increases significantly in the first 10 episodes, and then gradually reduces until convergence. The reason is that, at the beginning of the training process, the VMO might want to increase the user utility, so it requests many channels. However, this kind of action also makes the resource price increase at a faster rate. As a result, the VMO needs to adjust its requirements so that the rewards still increase and the cost is not too high. Finally, after about 150 episodes in the training process, the TDQL agent can learn the variations of the environment, and the VMO can guarantee the satisfaction of the users while not paying a high cost to the MNO.

In Figure 5.11, we show the average leasing cost of the VMOs from the effect of harvested energy. In this simulation, the average harvested energy at a base station was varied from 1 to 9 energy packets. From the figure, we can see that the leasing costs become much higher when the base stations receive more energy packets from their harvesting devices, which allows the BSs to transmit more data to the users. Therefore, the VMOs might request more resources from the MNO to serve their subscribers, which increases the

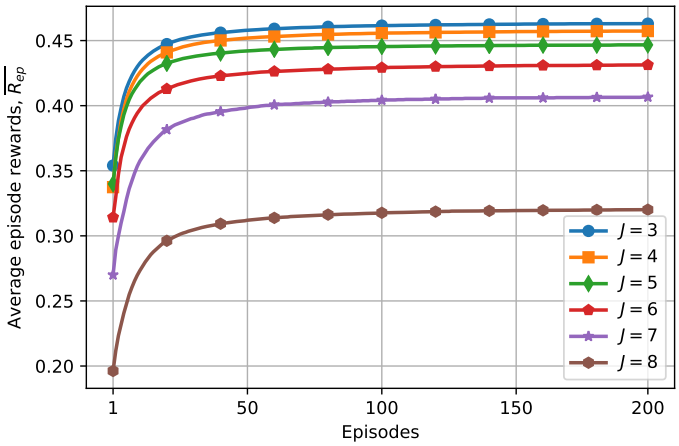

Figure 5.12: Convergence behavior of the proposed algorithm when the number of VMOs in the network changes.

unit price of the radio channels. Furthermore, the TDQL-based VMO pays a lower cost than the myopic-based VMO. This is because VMO 1 can predict the arrival of the harvested energy as well as the variations in the resource prices, and thus, makes a better decision in each time slot. To summarize, the proposed TDQL scheme can exploit the well-trained Q-network to train the agent in a new but similar environment. Hence, the rewards and the costs for the TDQL VMO are improved efficiently.

In the last experiment, we validated the performance of the proposed scheme when the number of VMOs in the network, J, was varied from 3 to 8, as shown in Figure 5.12 and Figure 5.13. In this experiment, the first three VMOs (i.e., VMOs 1, 2, and 3) use the same strategies as in previous simulations, whereas the remaining VMOs use the myopic scheme to compete for spectrum resources. We can observe from the figures that the increment in the number of VMOs does not affect the convergence speed of the proposed scheme. The TDQL agent can still learn the optimal resource-leasing policy after 100 training episodes. However, when J is too large (e.g., $J \geq 7$), the current amount of spectrum resources in the system might not be enough for the demands of the whole network, and thus, results in a dramatic decrease in the average reward of each VMO. In practical applications, it is more reasonable if the number of radio channels that are available for lease is also increased when the size of the whole network increases. As the amount of spectrum resources changes from $N = 15$ (channels) to $N = 20$ (channels), the VMOs can provide better performance to their subscribers, and thus, gain higher rewards. Once again, the proposed learning-

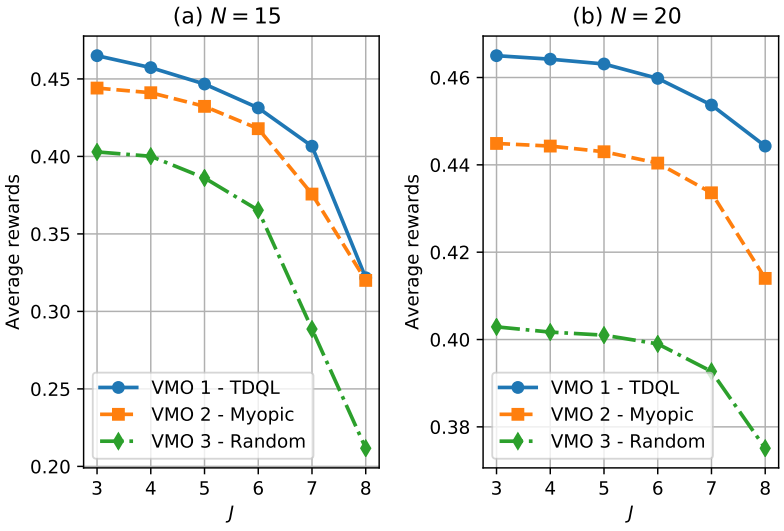

Figure 5.13: Average rewards for the VMOs based on the number of VMOs in the network with different values of N.

based approach outperforms the other methods for spectrum leasing in wireless virtualized networks.

5.5 Conclusion

In this paper, we consider a virtual mobile network in which several virtual mobile operators are leasing the radio channels from a mobile network operator. We propose a dynamic resource-leasing scheme based on a transfer deep–Q-learning algorithm, which allows the VMOs to compete for the radio resources for their users by learning the dynamics of harvested energy, resource prices, and data requests. We model the resource-leasing problem as the framework of a Markov decision process, during which the VMO tries to find the optimal announced resources to maximize utility. From the simulation results, the agent in the proposed approach can adapt its strategy to the variations in harvested energy, resource prices and demand from subscribers, and thus, achieves a greater reward than the others. Besides, we adopt the idea of transfer learning in our work to improve the learning speed of the agent by making use of a trained Q-network from the historical period.

Chapter 6

Summary of Contributions and Future Works

6.1 Introduction

Previous chapters have presented the research motivations, the problems, and solutions regarding the applications of artificial intelligence in wireless networks (e.g., for information security and resource management). This chapter summarizes the main contributions of this dissertation and discusses future research directions.

6.2 Summary of Contributions

This dissertation discusses the applications of artificial intelligence (AI) techniques, such as reinforcement learning and deep learning, in wireless communication networks, which aim to enhance the overall network performance. The main contributions of this dissertation are summarized as follows:

Firstly, we propose learning-based techniques for cooperative spectrum sensing (CSS) and energy-efficient data protection in cognitive radio networks (CRNs). We design a new convolutional neural network (CNN)-based CSS method that trains a CNN for spectrum sensing by using historical sensing data collected from secondary users (SUs) under various environmental conditions. The proposed CSS can increase detection probability and reduce sensing errors. We also propose two data protection schemes, based on which the SU determines its operation mode considering its remaining energy and the sensing

result to improve the security level of the transmitted data. The first scheme, namely the POMDP method, can provide the SU with the best performance. However, this scheme requires prior information about the environment dynamics, such as the energy harvesting model and the activity model of the primary user (PU). The second scheme, namely the transfer actor-critic learning algorithm, does not require information about the environment in advance. Instead, the learning agent can learn about those dynamics by interacting with the environment.

Secondly, we propose two energy-efficient power allocation schemes for data transmission against a full-duplex eavesdropper in a cognitive-aided wireless sensor network. In such a network, a sensor node (i.e., a source) want to protect the data sent to a cluster head (i.e., a destination) in the presence of an active eavesdropper. The source frequently performs spectrum sensing to detect the jamming activity of the eavesdropper in the network, and it then sends the local sensing result to the cluster head for making a global decision about the jamming state. Furthermore, the destination also interferes with the eavesdropping process by sending artificial noise against the eavesdropper. Based on the global sensing result, the source can effectively allocate power for data transmissions to maximize the long-term secrecy rate of the system under the constraint of harvested energy. The problem is first formulated and solved based on a POMDP framework (i.e., the first proposed scheme). We go further and propose an actor-critic learning framework to find the solution from practical interactions with the environment. The simulation results show that our proposed solutions can efficiently enhance data security and energy utilization in the long run.

Thirdly, we investigate a deep learning framework for joint user association and bandwidth allocation in dense mobile networks with energy-harvesting base stations. More specifically, we formulated the optimization problem (adhering to constraints on harvested energy and bandwidth) as a Markov decision process. We then employed an actor-critic algorithm to find the optimal solution for maximizing the system rewards. We further exploited deep neural networks to approximate the policy function and the value function, which allowed the algorithm to work with large state and action spaces. The agent of the ACDL algorithm can find the optimal policy through interactions with the environment. Consequently, the controller can effectively associate users with the base stations, and can then allocate bandwidth for their data transmissions based on the current state of the network. The simulation results show the advantage of the proposed solution in improving

network performance in the long run.

Finally, we consider a mobile network in which several virtual mobile operators (VMOs) are leasing the radio channels from a mobile network operator. We propose a dynamic resource-leasing scheme based on a transfer deep Q-learning algorithm, which allows the VMOs to compete for the radio resources for their users by learning the dynamics of harvested energy, resource prices, and data requests. We model the resource-leasing problem as the framework of a Markov decision process, during which a VMO tries to find the optimal announced resources to maximize utility. From the simulation results, the agent in the proposed approach can adapt its strategy to the variations in harvested energy, resource prices and demand from subscribers, and thus, achieves a greater reward than the others. Besides, we adopt the idea of transfer learning in our work to improve the learning speed of the agent by making use of a transferred Q-network that is well trained from the historical periods.

6.3 Future Works

For future research direction regarding artificial intelligence-based techniques (i.e., deep reinforcement learning algorithms) for efficient resource management in wireless networks, we consider several aspects as follows:

DRL for network access and power control

Modern networks, such as the Internet of Things, have become more decentralized. In such networks, entities need to make local decisions (e.g., user associations and base station selections) to achieve their own goals. This is challenging due to the dynamic and the uncertainty of the network status. Deep reinforcement learning algorithms allow network entities to build knowledge about the networks to make intelligent decisions. Thus, DRL can be used to solved the following issues: *dynamic spectrum access*, *joint user association and spectrum access*, and *power control*. However, the entities might not have sufficient observations about the system, e.g., channel states, base station capacity and energy, and system bandwidth. Therefore, DRL can be adopted to effectively solve the problems instead of using dynamic programming that requires high computational complexity and complete network information.

DRL for optimizing edge

Edge computing has become one of the key features in many information-centric networks since it can significantly reduce service latency, energy consumption, and cloud computing pressure. Joint content caching and offloading can address the gap between users' large data demands and the limited capacities of the network entities in terms of data storage and processing. However, deploying edge caching and edge computing (ECEC) in large-scale networks requires complicated system analysis due to stochastic features in user mobility, user demand, quality of service (QoS) provisioning, radio interfaces, and radio resources. For this reason, DRL-based approaches become a promising solution to optimization problems with large state and action spaces. More specifically, the DQL framework for caching can be implemented at the network controller, e.g., the base station, service provider, and central scheduler. Meanwhile, DRL for edge computing can be implemented at local devices, e.g., mobile users, IoT devices, and fog nodes.

DRL for cyber security

Internet-connected systems become more decentralized and ad-hoc in nature, and thus, are vulnerable to cyber-physical attacks more than ever. Recently, DQL techniques have been developed as an effective solution to avoid and prevent attacks. In cyber environments, DRL can be used not only for enhancing the communications and networking capabilities of IoT applications but also for defending against cyber attacks. Thus, DRL framework can be used to solve the following issues: *intrusion detecton*, *jamming attack*, *cyber-physical attack*, and *connectivity preserving*. Although DRL can help enhance the network security, the applications of DQL for the cyber-physical security are relatively few and need to be investigated. For example, the defender in cyber-physical systems can be represented by an actor-critic DRL agent that can learn the optimal strategy to timely and accurately defend the systems from unknown cyber-attacks. DRL algorithms can also be facilitated for handling or mitigating jamming attacks in edge networks by providing secure offloading to the edge nodes against jamming attacks. Furthermore, DRL can be an effective solution to intrusion detection problems by feeding the system data into a Markov process and predicting abnormal behaviors of the system.

Publications

Journals

[1] Quang Vinh Do and Insoo Koo, "A Transfer Deep Q-Learning Framework for Resource Competition in Virtual Mobile Networks with Energy-Harvesting Base Stations," *IEEE Systems Journal*, 2019.

[2] Quang Vinh Do and Insoo Koo, "Actor-critic Deep Learning for Efficient User Association and Bandwidth Allocation in Dense Mobile Networks with Green Base Stations," *Wireless Networks*, vol. 25, no. 8, pp. 5057-5068, Nov. 2019.

[3] Quang Vinh Do, T.N.K. Hoan, and Insoo Koo, "Optimal Power Allocation for Energy-efficient Data Transmission Against Full-duplex Active Eavesdroppers in Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 19, no. 13, pp. 5333-5346, July 2019.

[4] Quang Vinh Do, Van Hiep Vu, and Insoo Koo, "An Efficient Bandwidth Allocation Scheme for Hierarchical Cellular Networks with Energy Harvesting: An Actor-Critic Approach," *International Journal of Electronics*, vol. 106, no. 10, pp. 1543-1566, Apr. 2019.

[5] Quang Vinh Do and Insoo Koo, "Learning Frameworks for Cooperative Spectrum Sensing and Energy-Efficient Data Protection in Cognitive Radio Networks," *Applied Science*, vol. 8, no. 5, pp. 722-745, May 2018.

[6] Quang Vinh Do, T.N.K. Hoan, and Insoo Koo, "Energy-Efficient Data Encryption Scheme for Cognitive Radio Networks," *IEEE Sensors Journal*, vol. 18, no. 5, pp. 2050-2059, Mar. 2018.

[7] Quang Vinh Do and Insoo Koo, "FPGA Implementation of LSB-Based Steganography," *Journal of Information and Communication Convergence Engineering*, vol. 15, no. 3, pp. 151-159, 2017.

Papers under review

[8] Quang Vinh Do and Insoo Koo, "Dynamic Spectrum Leasing Based on Deep Reinforcement Learning in Cognitive Virtualized Networks with Green Base Stations," 2020.

Conferences

[9] Quang Vinh Do and Insoo Koo, "Dynamic Bandwidth Allocation Scheme for Wireless Networks with Energy Harvesting Using Actor-Critic Deep Reinforcement Learning," *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Jan. 2019, Okinawa, Japan.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. London: A Bradford Book, 2018.

[2] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[3] M. Song, C. Xin, Y. Zhao, and X. Cheng, "Dynamic spectrum access: from cognitive radio to network radio," *IEEE Wireless Communications*, vol. 19, no. 1, pp. 23–29, Feb. 2012.

[4] A. Ghasemi and E. S. Sousa, "Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 32–39, Apr. 2008.

[5] H. Zhang, S. Huang, C. Jiang, K. Long, V. C. M. Leung, and H. V. Poor, "Energy efficient user association and power allocation in millimeter-wave-based ultra dense networks with energy harvesting base stations," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 9, pp. 1936–1947, Sep. 2017.

[6] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud rans," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[7] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, Aug. 1999.

[8] M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu, and A. V. Vasilakos, "Software-defined and virtualized future mobile and wireless networks: A survey," *Mobile Networks and Applications*, vol. 20, no. 1, pp. 4–18, Feb. 2015.

[9] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 358–380, Firstquarter 2015.

[10] F. Granelli, A. A. Gebremariam, M. Usman, F. Cugini, V. Stamati, M. Alitska, and P. Chatzimisios, "Software defined and virtualized wireless access in future wireless networks: scenarios and standards," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 26–34, Jun. 2015.

[11] G. Joshi, S. Nam, and S. Kim, "Cognitive Radio Wireless Sensor Networks: Applications, Challenges and Research Trends," *Sensors*, vol. 13, no. 9, pp. 11 196–11 228, Aug. 2013.

[12] S. Park and D. Hong, "Optimal Spectrum Access for Energy Harvesting Cognitive Radio Networks," *IEEE Trans. Wirel. Commun.*, vol. 12, no. 12, pp. 6166–6179, Dec. 2013.

[13] S. Park, H. Kim, and D. Hong, "Cognitive radio networks with energy harvesting," *IEEE Trans. Wirel. Commun.*, vol. 12, no. 3, pp. 1386–1397, 2013.

[14] N. Pappas, J. Jeon, A. Ephremides, and A. Traganitis, "Optimal utilization of a cognitive shared channel with a rechargeable primary source node," *J. Commun. Networks*, vol. 14, no. 2, pp. 162–168, Apr. 2012.

[15] A. Sultan, "Sensing and Transmit Energy Optimization for an Energy Harvesting Cognitive Radio," *IEEE Wirel. Commun. Lett.*, vol. 1, no. 5, pp. 500–503, Oct. 2012.

[16] A. Razaque and K. M. Elleithy, "Energy-efficient boarder node medium access control protocol for wireless sensor networks," *Sensors*, vol. 14, no. 3, pp. 5074–5117, 2014.

[17] Ying-Chang Liang, Yonghong Zeng, E. Peh, and Anh Tuan Hoang, "Sensing-Throughput Tradeoff for Cognitive Radio Networks," *IEEE Trans. Wirel. Commun.*, vol. 7, no. 4, pp. 1326–1337, Apr. 2008.

[18] S. Lee, R. Zhang, and K. Huang, "Opportunistic Wireless Energy Harvesting in Cognitive Radio Networks," *IEEE Trans. Wirel. Commun.*, vol. 12, no. 9, pp. 4788–4799, Sep. 2013.

[19] P. S. Rossi, D. Ciuonzo, and G. Romano, "Orthogonality and cooperation in collaborative spectrum sensing through mimo decision fusion," *IEEE Trans. Wirel. Commun.*, vol. 12, no. 11, pp. 5826–5836, Nov. 2013.

[20] S. Holcomb and D. B. Rawat, "Recent security issues on cognitive radio networks: A survey," in *SoutheastCon 2016*. IEEE, Mar. 2016, pp. 1–6.

[21] A. G. Fragkiadakis, E. Z. Tragos, and I. G. Askoxylakis, "A Survey on Security Threats and Detection Techniques in Cognitive Radio Networks," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 1, pp. 428–445, 2013.

[22] Hong Wen, Shaoqian Li, Xiping Zhu, and Liang Zhou, "A framework of the PHY-layer approach to defense against security threats in cognitive radio networks," *IEEE Netw.*, vol. 27, no. 3, pp. 34–39, May 2013.

[23] D. Ciuonzo, A. Aubry, and V. Carotenuto, "Rician mimo channel- and jamming-aware decision fusion," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 3866–3880, Aug. 2017.

[24] X. Xu, B. He, W. Yang, X. Zhou, and Y. Cai, "Secure Transmission Design for Cognitive Radio Networks With Poisson Distributed Eavesdroppers," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 2, pp. 373–387, Feb. 2016.

[25] M. Elkashlan, L. Wang, T. Q. Duong, G. K. Karagiannidis, and A. Nallanathan, "On the Security of Cognitive Radio Networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 8, pp. 3790–3795, Aug. 2015.

[26] B. Wang, Y. Zhan, and Z. Zhang, "Cryptanalysis of a Symmetric Fully Homomorphic Encryption Scheme," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 6, pp. 1460–1467, 2018.

[27] A. Sultan, X. Yang, A. A. Hajomer, and W. Hu, "Chaotic Constellation Mapping for Physical-Layer Data Encryption in OFDM-PON," *IEEE Photonics Technol. Lett.*, vol. 30, no. 4, pp. 339–342, 2018.

[28] S. Angizi, Z. He, N. Bagherzadeh, and D. Fan, "Design and Evaluation of a Spintronic In-Memory Processing Platform for Non-Volatile Data Encryption," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, 2017.

[29] J. Sen, "A Survey on Security and Privacy Protocols for Cognitive Wireless Sensor Networks," *J. Netw. Inf. Secur.*, vol. 1, pp. 1–43, 2013.

[30] J. M. Kim, H. S. Lee, J. Yi, and M. Park, "Power Adaptive Data Encryption for Energy-Efficient and Secure Communication in Solar-Powered Wireless Sensor Networks," *Journal of Sensors*, 2016.

[31] NIST Standards, "Advanced encryption standard (AES)," 2001. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

[32] D. J. Bernstein, "Understanding brute force," *ECRYPT STVL Workshop on Symmetric Key Encryption*, pp. 10–19, 2005.

[33] H. Berenji and D. Vengerov, "A convergent actor-critic-based FRL algorithm with application to power management of wireless transmitters," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 4, pp. 478–485, Aug. 2003.

[34] M. E. Taylor and P. Stone, "Transfer Learning for Reinforcement Learning Domains: A Survey," *J. Mach. Learn. Res.*, vol. 10, pp. 1633–1685, 2009.

[35] P. Lee, Z. A. Eu, M. Han, and H.-P. Tan, "Empirical modeling of a solar-powered energy harvesting wireless sensor node for time-slotted operation," in *2011 IEEE Wirel. Commun. Netw. Conf.* IEEE, Mar. 2011, pp. 179–184.

[36] S. Zhang, H. Wang, and X. Zhang, "Estimation of channel state transition probabilities based on Markov Chains in cognitive radio," *J. Commun.*, vol. 9, no. 6, pp. 468–474, 2014.

[37] S. B. Othman, "Performance evaluation of encryption algorithm for wireless sensor networks," *Inf. Technol. e-Services (ICITeS), 2012 Int. Conf.*, pp. 1 – 8, 2012.

[38] W. Zhang, R. Mallik, and K. Letaief, "Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks," *IEEE Trans. Wirel. Commun.*, vol. 8, no. 12, pp. 5761–5766, Dec. 2009.

[39] S. Atapattu, C. Tellambura, and H. Jiang, "Conventional Energy Detector," in *Energy Detect. Spectr. Sens. Cogn. Radio*, 1st ed. New York: Springer-Verlag, 2014, ch. 2, pp. 11–26.

[40] Z. Quan, S. Cui, and A. H. Sayed, "Optimal Linear Cooperation for Spectrum Sensing in Cognitive Radio Networks," *IEEE J. Sel. Top. Signal Process.*, vol. 2, no. 1, pp. 28–40, Feb. 2008.

[41] A. Gulli and S. Pal, *Deep Learning with Keras.* Birmingham: Packt Publishing Ltd, 2017.

[42] D. P. Bertsekas, *Dynamic Programming and Optimal Control.* Athena Scientific Belmont, MA, 2007, vol. I.

[43] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Mach. Learn.*, vol. 38, no. 3, pp. 287–308, 2000.

[44] D. Teguig, B. Scheers, and V. L. Nir, "Data fusion schemes for cooperative spectrum sensing in cognitive radio networks," in *2012 Mil. Commun. Inf. Syst. Conf.*, 2012, pp. 1–7.

[45] H. Vu-Van and I. Koo, "Cooperative spectrum sensing with collaborative users using individual sensing credibility for cognitive radio network," *IEEE Trans. Consum. Electron.*, vol. 57, no. 2, pp. 320–326, 2011.

[46] L. C. Li, J. Wang, and S., "An adaptive cooperative spectrum sensing scheme based on the optimal data fusion rule," in *4th Int. Symp. Wirel. Commun. Syst.*, Oct. 2007, pp. 582–586.

[47] Q. Zhao, B. Krishnamachari, and K. Liu, "On myopic sensing for multi-channel opportunistic access: structure, optimality, and performance," *IEEE Trans. Wirel. Commun.*, vol. 7, no. 12, pp. 5431–5440, Dec. 2008.

[48] D. Culler, D. Estrin, and M. Srivastava, "Guest Editors' Introduction: Overview of Sensor Networks," *Computer*, vol. 37, no. 8, pp. 41–49, Aug. 2004.

[49] Sainath Gopi Nambiar and P. Ranjan, "Energy harvesting system for deployment of Wireless Sensor Networks in Nuclear Fusion Reactor," in *2012 International Conference on Green Technologies (ICGT).* IEEE, Dec. 2012, pp. 288–292.

[50] D. Dondi, A. Bertacchini, D. Brunelli, L. Larcher, and L. Benini, "Modeling and Optimization of a Solar Energy Harvester System for Self-Powered Wireless Sensor Networks," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 7, pp. 2759–2766, Jul. 2008.

[51] Yen Kheng Tan and S. K. Panda, "Optimized Wind Energy Harvesting System Using Resistance Emulator and Active Rectifier for Wireless Sensor Nodes," *IEEE Transactions on Power Electronics*, vol. 26, no. 1, pp. 38–50, Jan. 2011.

[52] A. Cuadras, M. Gasulla, and V. Ferrari, "Thermal energy harvesting through pyroelectricity," *Sensors and Actuators A: Physical*, vol. 158, no. 1, pp. 132–139, Mar. 2010.

[53] C. Saha, T. O'Donnell, H. Loder, S. Beeby, and J. Tudor, "Optimization of an Electromagnetic Energy Harvesting Device," *IEEE Transactions on Magnetics*, vol. 42, no. 10, pp. 3509–3511, Oct. 2006.

[54] A. C. Valera, W.-S. Soh, and H.-P. Tan, "Survey on wakeup scheduling for environmentally powered wireless sensor networks," *Computer Communications*, vol. 52, pp. 21–36, Oct. 2014.

[55] F. Akhtar and M. H. Rehmani, "Energy replenishment using renewable and traditional energy resources for sustainable wireless sensor networks: A review," *Renewable and Sustainable Energy Reviews*, vol. 45, pp. 769–784, 2015.

[56] L.-G. Tran, H.-K. Cha, and W.-T. Park, "RF power harvesting: a review on designing methodologies and applications," *Micro and Nano Systems Letters*, vol. 5, no. 1, p. 14, 2017.

[57] F. K. Shaikh and S. Zeadally, "Energy harvesting in wireless sensor networks: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 1041–1054, Mar. 2016.

[58] A. Collado and A. Georgiadis, "Conformal Hybrid Solar and Electromagnetic (EM) Energy Harvesting Rectenna," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 8, pp. 2225–2234, Aug. 2013.

[59] N. Yang, L. Wang, G. Geraci, M. Elkashlan, J. Yuan, and M. D. Renzo, "Safeguarding 5G wireless communication networks using physical layer security," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 20–27, Apr. 2015.

[60] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, Sep. 2016.

[61] Y.-S. Shiu, S. Chang, H.-C. Wu, S. Huang, and H.-H. Chen, "Physical layer security in wireless networks: a tutorial," *IEEE Wireless Communications*, vol. 18, no. 2, pp. 66–74, Apr. 2011.

[62] M. Duarte, C. Dick, and A. Sabharwal, "Experiment-Driven Characterization of Full-Duplex Wireless Systems," *IEEE Transactions on Wireless Communications*, vol. 11, no. 12, pp. 4296–4307, Dec. 2012.

[63] G. Liu, F. R. Yu, H. Ji, V. C. M. Leung, and X. Li, "In-Band Full-Duplex Relaying: A Survey, Research Issues and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 500–524, 2015.

[64] W. Li, M. Ghogho, B. Chen, and C. Xiong, "Secure Communication via Sending Artificial Noise by the Receiver: Outage Secrecy Capacity/Region Analysis," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1628–1631, Oct. 2012.

[65] L. Chen, Q. Zhu, W. Meng, and Y. Hua, "Fast Power Allocation for Secure Communication With Full-Duplex Radio," *IEEE Transactions on Signal Processing*, vol. 65, no. 14, pp. 3846–3861, Jul. 2017.

[66] X. Zhou, B. Maham, and A. Hjorungnes, "Pilot Contamination for Active Eavesdropping," *IEEE Transactions on Wireless Communications*, vol. 11, no. 3, pp. 903–907, Mar. 2012.

[67] A. Al-nahari, "Physical layer security using massive multiple-input and multiple-output: passive and active eavesdroppers," *IET Communications*, vol. 10, no. 1, pp. 50–56, Jan. 2016.

[68] Y. Wu, R. Schober, D. W. K. Ng, C. Xiao, and G. Caire, "Secure Massive MIMO Transmission With an Active Eavesdropper," *IEEE Transactions on Information Theory*, vol. 62, no. 7, pp. 3880–3900, Jul. 2016.

[69] G. Vijay, E. Ben Ali Bdira, and M. Ibnkahla, "Cognition in Wireless Sensor Networks: A Perspective," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 582–592, Mar. 2011.

[70] A. O. Bicen, V. C. Gungor, and O. B. Akan, "Delay-sensitive and multimedia communication in cognitive radio sensor networks," *Ad Hoc Networks*, vol. 10, no. 5, pp. 816–830, Jul. 2012.

[71] Y.-W. P. Hong, P.-C. Lan, and C.-C. J. Kuo, "Enhancing Physical-Layer Secrecy in Multiantenna Wireless Systems: An Overview of Signal Processing Approaches," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 29–40, Sep. 2013.

[72] P.-H. Lin, S.-H. Lai, S.-C. Lin, and H.-J. Su, "On Secrecy Rate of the Generalized Artificial-Noise Assisted Secure Beamforming for Wiretap Channels," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 1728–1740, Sep. 2013.

[73] Q. Li, Y. Yang, W.-K. Ma, M. Lin, J. Ge, and J. Lin, "Robust Cooperative Beamforming and Artificial Noise Design for Physical-Layer Secrecy in AF Multi-Antenna Multi-Relay Networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 1, pp. 206–220, Jan. 2015.

[74] Y. Zou, X. Wang, and W. Shen, "Intercept probability analysis of cooperative wireless networks with best relay selection in the presence of eavesdropping attack," in *2013 IEEE International Conference on Communications (ICC)*. IEEE, Jun. 2013, pp. 2183–2187.

[75] G. Zheng, I. Krikidis, J. Li, A. P. Petropulu, and B. Ottersten, "Improving Physical Layer Secrecy Using Full-Duplex Jamming Receivers," *IEEE Transactions on Signal Processing*, vol. 61, no. 20, pp. 4962–4974, Oct. 2013.

[76] J. Qiao, H. Zhang, D. Wu, and D. Yuan, "Secrecy rate analysis for jamming assisted relay communications systems," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2015-Augus. IEEE, Apr. 2015, pp. 3143–3147.

[77] A. Mukherjee and A. L. Swindlehurst, "A full-duplex active eavesdropper in mimo wiretap channels: Construction and countermeasures," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*. IEEE, Nov. 2011, pp. 265–269.

[78] X. Tang, P. Ren, Y. Wang, and Z. Han, "Combating Full-Duplex Active Eavesdropper: A Hierarchical Game Perspective," *IEEE Transactions on Communications*, vol. 65, no. 3, pp. 1379–1395, Mar. 2017.

[79] D. T. Hoang, D. Niyato, P. Wang, and D. I. Kim, "Performance Optimization for Cooperative Multiuser Cognitive Radio Networks with RF Energy Harvesting Capability," *IEEE Transactions on Wireless Communications*, vol. 14, no. 7, pp. 3614–3629, Jul. 2015.

[80] D. Niyato, P. Wang, D. I. Kim, W. Saad, and Z. Han, "Mobile Energy Sharing Networks: Performance Analysis and Optimization," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3519–3535, May 2016.

[81] A. Asheralieva, "Bayesian Reinforcement Learning-Based Coalition Formation for Distributed Resource Sharing by Device-to-Device Users in Heterogeneous Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 5016–5032, Aug. 2017.

[82] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User Scheduling and Resource Allocation in HetNets With Hybrid Energy Supply: An Actor-Critic Reinforcement Learning Approach," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 680–692, Jan. 2018.

[83] P. Kuila and P. K. Jana, "Energy efficient clustering and routing algorithms for wireless sensor networks: Particle swarm optimization approach," *Engineering Applications of Artificial Intelligence*, vol. 33, pp. 127–140, Aug. 2014.

[84] Q. V. Do and I. Koo, "Learning Frameworks for Cooperative Spectrum Sensing and Energy-Efficient Data Protection in Cognitive Radio Networks," *Applied Sciences*, vol. 8, no. 5, p. 722, May 2018.

[85] Jun Ma, Guodong Zhao, and Ye Li, "Soft Combination and Detection for Cooperative Spectrum Sensing in Cognitive Radio Networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 11, pp. 4502–4507, Nov. 2008.

[86] K. Cichon, A. Kliks, and H. Bogucka, "Energy-Efficient Cooperative Spectrum Sensing: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1861–1886, 2016.

[87] T. Le, K. Mayaram, and T. Fiez, "Efficient Far-Field Radio Frequency Energy Harvesting for Passively Powered Sensor Networks," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 5, pp. 1287–1302, May 2008.

[88] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge: Cambridge University Press, 2005.

[89] X.-R. Cao and X. Guo, "Partially Observable Markov Decision Processes With Reward Information: Basic Ideas and Models," *IEEE Transactions on Automatic Control*, vol. 52, no. 4, pp. 677–681, Apr. 2007.

[90] W. Wang, A. Kwasinski, D. Niyato, and Z. Han, "A Survey on Applications of Model-Free Strategy Learning in Cognitive Wireless Networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1717–1757, 2016.

[91] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems 12*, S. A. Solla, T. K. Leen, and K. Müller, Eds. MIT Press, 2000, pp. 1008–1014.

[92] F. Tian, X. Chen, S. Liu, X. Yuan, D. Li, X. Zhang, and Z. Yang, "Secrecy Rate Optimization in Wireless Multi-Hop Full Duplex Networks," *IEEE Access*, vol. 6, pp. 5695–5704, 2018.

[93] A. Al-Talabani, Y. Deng, A. Nallanathan, and H. X. Nguyen, "Enhancing Secrecy Rate in Cognitive Radio Networks via Stackelberg Game," *IEEE Transactions on Communications*, vol. 64, no. 11, pp. 4764–4775, Nov. 2016.

[94] Z. Gao, L. Dai, D. Mi, Z. Wang, M. A. Imran, and M. Z. Shakir, "Mmwave massive-mimo-based wireless backhaul for the 5g ultra-dense network," *IEEE Wireless Communications*, vol. 22, no. 5, pp. 13–21, Oct. 2015.

[95] Chun-Ting Chou and K. G. Shin, "Analysis of adaptive bandwidth allocation in wireless networks with multilevel degradable quality of service," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 5–17, Jan. 2004.

[96] Z. Zhou, M. Dong, K. Ota, G. Wang, and L. T. Yang, "Energy-efficient resource allocation for d2d communications underlaying cloud-ran-based lte-a networks," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 428–438, Jun. 2016.

[97] Yuhan Liu and Jian Su, "Priority-based bandwidth allocation in heterogeneous wireless network," in *11th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2015)*, Sep. 2015, pp. 1–6.

[98] A. Esmailpour and N. Nasser, "Dynamic qos-based bandwidth allocation framework for broadband wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 6, pp. 2690–2700, Jul. 2011.

[99] T. Han and N. Ansari, "On optimizing green energy utilization for cellular networks with hybrid energy supplies," *IEEE Transactions on Wireless Communications*, vol. 12, no. 8, pp. 3872–3882, Aug. 2013.

[100] H. S. Dhillon, Y. Li, P. Nuggehalli, Z. Pi, and J. G. Andrews, "Fundamentals of heterogeneous cellular networks with energy harvesting," *IEEE Transactions on Wireless Communications*, vol. 13, no. 5, pp. 2782–2797, May 2014.

[101] Z. Hasan, H. Boostanimehr, and V. K. Bhargava, "Green cellular networks: A survey, some research issues and challenges," *IEEE Communications Surveys Tutorials*, vol. 13, no. 4, pp. 524–540, Fourth Quarter 2011.

[102] K. Son, S. Chong, and G. D. Veciana, "Dynamic association for load balancing and interference avoidance in multi-cell networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 7, pp. 3566–3576, Jul. 2009.

[103] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.

[104] J. Andrews, S. Singh, Q. Ye, X. Lin, and H. Dhillon, "An overview of load balancing in hetnets: old myths and open problems," *IEEE Wireless Communications*, vol. 21, no. 2, pp. 18–25, Apr. 2014.

[105] R. Xie, F. R. Yu, H. Ji, and Y. Li, "Energy-efficient resource allocation for heterogeneous cognitive radio networks with femtocells," *IEEE Transactions on Wireless Communications*, vol. 11, no. 11, pp. 3910–3920, Nov. 2012.

[106] C. M. Gabriel Gussen, E. V. Belmega, and M. Debbah, "Pricing and bandwidth allocation problems in wireless multi-tier networks," in *Proceedings of The 2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*. IEEE, Nov. 2011, pp. 1633–1637.

[107] R. Li, Z. Zhao, X. Chen, J. Palicot, and H. Zhang, "TACT: A Transfer Actor-Critic Learning Framework for Energy Saving in Cellular Radio Access Networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 4, pp. 2000–2011, Apr. 2014.

[108] S. Spielberg, R. Gopaluni, and P. Loewen, "Deep reinforcement learning approaches for process control," in *2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP)*. IEEE, May 2017, pp. 201–206.

[109] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," in *Proceedings of The 2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–7.

[110] Wenye Wang, Xinbing Wang, and A. A. Nilsson, "Energy-efficient bandwidth allocation in wireless networks: algorithms, analysis, and simulations," *IEEE Transactions on Wireless Communications*, vol. 5, no. 5, pp. 1103–1114, May 2006.

[111] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.

[112] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning,"

in *Proceeding of The 33rd International Conference on International Conference on Machine Learning*, New York, NY, USA, 2016, pp. 1928–1937.

[113] 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," in *Proceedings of The International Conference on Learning Representations*, San Juan, Puerto Rico, 2016, pp. 1–14.

[114] Hado van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proceedings of The Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, USA, 2016, pp. 2094–2100.

[115] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[116] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceeding of The Thirteenth International Conference on Artificial Intelligence and Statistics*. Sardinia, Italy: PMLR, 2010, pp. 249–256.

[117] K. Wang, L. Chen, and Q. Liu, "On Optimality of Myopic Policy for Opportunistic Access With Nonidentical Channels and Imperfect Sensing," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2478–2483, Jun. 2014.

[118] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of The 3rd International Conference for Learning Representations*, San Diego, 2015, pp. 1–15.

[119] C. I, C. Rowell, S. Han, Z. Xu, G. Li, and Z. Pan, "Toward green and soft: A 5G perspective," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 66–73, Feb. 2014.

[120] G. Piro, M. Miozzo, G. Forte, N. Baldo, L. A. Grieco, G. Boggia, and P. Dini, "HetNets powered by renewable energy sources: Sustainable next-generation cellular networks," *IEEE Internet Computing*, vol. 17, no. 1, pp. 32–39, Jan. 2013.

[121] G. Sun, Z. T. Gebrekidan, G. O. Boateng, D. Ayepah-Mensah, and W. Jiang, "Dynamic reservation and deep reinforcement learning based autonomous resource slicing for virtualized radio access networks," *IEEE Access*, vol. 7, pp. 45 758–45 772, 2019.

[122] C. Liang and F. R. Yu, "Wireless virtualization for next generation mobile cellular networks," *IEEE Wireless Communications*, vol. 22, no. 1, pp. 61–69, Feb. 2015.

[123] M. Derakhshani, S. Parsaeefard, T. Le-Ngoc, and A. Leon-Garcia, "Leveraging synergy of SDWN and multi-layer resource management for 5G networks," *IET Networks*, vol. 7, no. 5, pp. 336–345, 2018.

[124] X. Zhang and Q. Zhu, "Information-centric network virtualization for QoS provisioning over software defined wireless networks," in *MILCOM 2016 - 2016 IEEE Military Communications Conference*, Nov. 2016, pp. 1028–1033.

[125] Q. Han, B. Yang, G. Miao, C. Chen, X. Wang, and X. Guan, "Backhaul-aware user association and resource allocation for energy-constrained HetNets," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 580–593, Jan. 2017.

[126] L. Chen, F. R. Yu, H. Ji, G. Liu, and V. C. M. Leung, "Distributed virtual resource allocation in small-cell networks with full-duplex self-backhauls and virtualization," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 7, pp. 5410–5423, Jul. 2016.

[127] C. Wu, R. Wang, P. Wang, Y. Cao, L. Liu, K. Zhu, and B. Chen, "On the profit maximization of spectrum investment under uncertainties in cognitive radio networks," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[128] F. Fu and U. C. Kozat, "Stochastic game for wireless network virtualization," *IEEE/ACM Transactions on Networking*, vol. 21, no. 1, pp. 84–97, Feb. 2013.

[129] K. Guo, C. Yang, and T. Liu, "Caching in base station with recommendation via Q-Learning," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2017, pp. 1–6.

[130] T. Z. Oo, N. H. Tran, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Offloading in HetNet: A coordination of interference mitigation, user association, and resource allocation," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2276–2291, Aug. 2017.

[131] M. Simsek, M. Bennis, and . Güvenç, "Learning based frequency- and time-domain inter-cell interference coordination in HetNets," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4589–4602, Oct. 2015.

[132] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 310–323, Jan. 2019.

[133] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 257–265, Jun. 2018.

[134] R. Mijumbi, J. Gorricho, J. Serrat, M. Claeys, J. Famaey, and F. De Turck, "Neural network-based autonomous allocation of resources in virtual networks," in *2014 European Conference on Networks and Communications (EuCNC)*, Jun. 2014, pp. 1–6.

[135] R. Mijumbi, J.-L. Gorricho, J. Serrat, M. Shen, K. Xu, and K. Yang, "A neuro-fuzzy approach to self-management of virtual network resources," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1376 – 1390, 2015.

[136] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[137] R. Li, Z. Zhao, Q. Sun, C. I, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74 429–74 441, 2018.

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

[139] Y. Wei, F. R. Yu, M. Song, and Z. Han, "Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor–critic deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2061–2073, Apr. 2019.

[140] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *Proceedings of 2016 International Conference on Learning Representations*, San Juan, Puerto Rico, 2016, pp. 1–21.

[141] Y. He, Z. Zhang, F. R. Yu, N. Zhao, H. Yin, V. C. M. Leung, and Y. Zhang, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 433–10 445, Nov. 2017.