AI-BASED RADIO RESOURCE MANAGEMENT FOR ADVANCED WIRELESS COMMUNICATION NETWORKS

DISSERTATION

for the Degree of

DOCTOR OF PHILOSOPHY
(Electrical, Electronic and Computer Engineering)

HUYNH THANH THIEN

MAY 2022

AI-based Radio Resource Management for Advanced Wireless Communication Networks

Supervisor: Professor In-Soo Koo

DISSERTATION

Submitted in Partial Fulfillment
of the Requirements for the
Degree of

DOCTOR OF PHILOSOPHY
(Electrical, Electronic and Computer Engineering)

at the

UNIVERSITY OF ULSAN

by

Huynh Thanh Thien
May 2022

Publication No. ______________

AI-based Radio Resource Management for Advanced Wireless Communication Networks

Approved by Supervisory Committee:

Sungoh Kwon

Prof. Sungoh Kwon, Chair

InSoo Koo 구인

Prof. In-Soo Koo, Supervisor

Prof. YoungTae Noh

김명균

Prof. Myung-Kyun Kim

이치호

Dr. Chiho Lee

Department of Electrical, Electronic and Computer Engineering

University of Ulsan, South Korea

Date: May, 2022

VITA

Huynh Thanh Thien was born in Tra Vinh province, Vietnam, in 1989. He received a B.E. degree in Electronics and Telecommunications Engineering from Ton Duc Thang University, Ho Chi Minh, Vietnam, in 2012, and received an M.S. degree in Digital Mechatronic Technology, College of Engineering, from Chinese Culture University, Taipei, Taiwan, in 2014. Since February 2014, he has been working as a lecturer at the Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Vietnam.

Since February 2017, he has been pursuing a Ph.D. degree in Electrical, Electronic and Computer Engineering at the University of Ulsan (UOU), Ulsan, South Korea, under the supervision of Professor In-Soo Koo. His research interests include cognitive radio and next-generation wireless communications systems, game theory, optimization, deep learning, and reinforcement learning.

Dedicated to my dearest family and friends

for

their endless love, support and encouragement

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Prof. *In-Soo Koo* for all the support he gave me, during both the long months I spent undertaking my study in Korea and also the time I spent at the University of Ulsan. His kindness, patience, motivation, enthusiasm, constant support, and immense knowledge helped me a lot in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study.

Besides my advisor, I would like to thank the rest of my Ph.D. supervisory committee: Prof. *Sungoh Kwon*, Prof. *YoungTae Noh*, Prof. *Myung-Kyun Kim*, and Dr. *Chiho Lee*, for their encouragement, insightful comments, and hard questions that help a lot to improve the quality of this dissertation.

My sincere thanks also goes to Dr. *Vu Van Hiep*, and Dr. *Pham Viet Tuan*, for their valuable discussion, collaboration, and useful guidance throughout my Ph.D. study.

I am grateful to all members of the multimedia communications system laboratory (MCSL) for the stimulating discussions, their friendship, enthusiastic help, and cheerfulness, and for all the fun we have had. In addition, I am grateful to my friends, *Linh*, for her kind support, whenever I am in trouble to deal with any problem related to Korean interpretation, and *Thanh*, *Giang*, *Carla*, *Mario*, *Dung*, and *Toan* for all that we spent together.

I gratefully appreciate the BK21 Plus for financial support during my study in University of Ulsan.

Last but not least, I would like to thank my family for their spiritual support and encouragement throughout my life, especially my parents *Huynh Thanh Liem* and *Huynh Thi Huong*, and younger sister, *Huynh Thanh Nhi*, who have always supported every endeavor of mine.

Huynh Thanh Thien

Ulsan, South Korea, May - 2022.

ABSTRACT

AI-based Radio Resource Management for Advanced Wireless Communication Networks

by

Huynh Thanh Thien

Supervisor: Prof. In-Soo Koo

Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy (Electrical, Electronic and Computer Engineering)

May 2022

Nowadays, wireless communications systems have grown significantly. Therefore, using the radio frequency (RF) spectrum is increasing more dramatically in order to meet the growth in applications requiring broadband. Because of the spectrum scarcity, new spectrum management models are being developed to opportunistically utilize the dynamic spectrum access. Currently, the cognitive radio network (CRN) was developed and is considered one of the most promising technologies for improving spectrum efficiency. Cognitive radio (CR) was recognized as an enabling technology to mitigate the abuse of scarce RF spectrum in which dynamic spectrum access was proposed to share the available spectrum through opportunistic usage of the frequency bands by secondary operators without interfering with the primary networks. When CRN is installed, it enables secondary networks to perform the following tasks: spectrum sensing, spectrum management, and mobility management. CR technology offers the opportunity to optimize spectrum access, assists in preventing interference, and adapts to instant spectrum slot availability from the unused spectrum pool.

Along with rapid developments of mobile applications as well as expanding network infrastructure (transmission lines, terminal equipment, and base stations), efficient energy management is becoming an issue that deserves special attention. Efficient energy management helps to overcome the bottleneck of wireless network applications operating under battery and energy constraints. It not only helps to reduce a device's dependence on battery power and power consumption, but also provides a continuous power source for the long-term operation of devices on the network. As a result, wireless communications powered by external harvested energy and the simultaneous wireless information and power transfer (SWIPT) transmission have become promising techniques to solve the energy-constrained problem. Regarding external harvested energy, radio frequency-energy harvesting (RF-EH) is a potential solution for energy-constrained issue in wireless networking, where the wireless devices can harvest energy from ambient RF signals. Along with RF-EH, non-RF energy resources (e.g., solar, wind, etc.) can also provide perpetual energy and higher power density for rechargeable batteries of wireless users. Regarding the SWIPT system, both information and energy of the common transmit signal are transmitted to the receivers. Therefore, the received signal can be used for energy harvesting (EH) and information decoding (ID).

In addition, artificial intelligence (AI), which is defined as any process or device that perceives its environment and takes actions that maximize the chances of success for some predefined goal, is a feasible solution for the emerging complex communication system design. The recent advances in reinforcement learning (RL) and deep learning (DL) hold significant promise for solving very complex problems considered intractable until now. It is now appropriate to apply AI technology to advanced wireless communication networks (WCNs) to tackle optimized complicated decision making, physical layer design, network management and resource optimization tasks in such networks. In the study of wireless technologies and communication systems, AI will be a powerful tool and hot research topic with many potential application areas, e.g., channel modeling, wireless signal processing, and resource management. Motivated by the aforementioned survey, this dissertation will focus on these remaining issues about applying AI to radio resource management for advanced WCNs, such as CRN and SWIPT, as follows:

Firstly, in CRN, secondary users (SUs) are able to sense the absence of primary users (PUs) in the spectrum. Then, SUs use this information to opportunistically access the licensed spectrum. In this work, we utilize a software-defined radio testbed of energy detection (ED)-based spectrum sensing. The testbed was built based on the GNU's Not Unix (GNU) Radio software platform and Universal Software Radio Peripheral (USRP) National Instruments 2900 devices. In this case, a new block of energy detection is developed by using an out-of-tree module from GNU Radio. To successfully integrate CR into the cloud computing paradigm, we also implement cloud computing-based spectrum sensing by utilizing a cloud server with ThingSpeak, such that we can store, process, and share the sensing information more efficiently in a centralized way in the cloud server. In addition, we also present an implementation of real-time video transmission with spectrum-sensing among two USRP devices. In this work, spectrum-sensing is implemented at both transmitter and receiver. The transmitter senses the channel, and if the channel is free, a video signal (which could be a real-time signal from a video file) will be modulated and processed by GNU Radio and transmitted using a USRP. A USRP receiver also senses the channel, but in contrast, if the channel is busy, the signal is demodulated to reproduce the transmitted video signal. These works brings in several challenges, like spectrum-sensing in the devices' environment, and packets getting lost or corrupted over the air.

Secondly, although a CRN is a novel solution that promises to solve the spectrum scarcity problem and enhance spectrum utilization, unsecured CRN can easily be manipulated in order to attack legacy users on the communication channel. As a result, the network's performance significantly degrades. Therefore, communication channel security is an important issue that needs to be addressed in a CRN. In this work, we focus on improving the security of multi-channel communication in a CRN, while various jammers try to access channels of interest to prevent SUs from using them. By using game-theoretic concepts and by defining states, actions, and players' rewards, we propose game-based schemes that find the best channel for the SUs in order to avoid jammer's attacks on communication channels. Accordingly, the problem is finding the optimal channel to maximize the long-term reward of the SU where communication channels are not used by the PUs and are not jammed by attackers. In addition, the idea of transfer learning might be applied to the problem under consideration, and thus, a transfer Game-Actor-Critic (TGACT) scheme is proposed, which uses the transferred knowledge in a double-game period to accelerate the learning process and provide performance improvement in channel selection. The simulation results show that the proposed schemes are quite resistant to jammer attacks, and achieve better performance compared to other channel selection schemes.

Thirdly, the SWIPT systems can supply efficiently throughput and energy, have emerged as a potential research area in fifth-generation (5G) system. In this work, We investigate the SWIPT system with multi-user, single-input single-output (SISO) system. First, we solve the transmit power optimization problem, which provides the optimal strategy for getting minimum power while satisfying sufficient signal-to-interference-plus-noise ratio (SINR) and harvested energy requirements to ensure receiver circuits work in SWIPT systems where receivers are equipped with a power-splitting (PS) structure. Although optimization algorithms are able to achieve relatively high performance, they often entail a significant number of iterations, which raises many issues in computation costs and time for real-time applications. Therefore, we aim at providing a DL-based approach, which is a promising solution to address this challenging issue. DL architectures used include a type of Deep Neural Network (DNN): the Feed-Forward Neural Network (FFNN) and three types of Recurrent Neural Network (RNN): the Layer Recurrent Network (LRN), the Nonlinear AutoRegressive network with eXogenous inputs (NARX), and Long Short-Term Memory (LSTM). Through simulations, we show that the DL-based approach can approximate a complex optimization algorithm that optimizes transmit power in SWIPT systems with much less computation time.

Then, the demand for spectral and energy efficiency has significantly been increased along with new breakthroughs in programmable meta-material techniques. The integration of an intelligent reflecting surface (IRS) into the SWIPT systems has attracted much attention from operators in advanced WCNs such as 5G and sixth-generation (6G) networks. In addition, an IRS-assisted SWIPT system faces many security risks that can easily be compromised by eavesdroppers. In this work, we investigate the physical-layer secure and transmission optimization problem in an IRS-assisted SWIPT system where a PS scheme is installed in the user equipment (UE). In particular, our purpose is to maximize the system secrecy rate by jointly finding optimal solutions for transmitter power, PS factor of UE, and phase shifts matrix of IRS under the required minimum harvested energy and maximum transmitter power. We propose the alternating optimization (AO)-based scheme to obtain optimal solutions. The proposed AO-based scheme can effectively solve both convex and non-convex problems; however, applying them in practice still poses some difficulties due to the complexity and long computation time. This is because many mathematical transformations are used and the optimal solution needs a number of iterations to achieve convergence. Therefore, we also propose 5 types of data and DNN structures to potentially achieve efficiency in computations by using a DL-based approach. The simulation results indicate that the proposed IRS scheme provides an improvement in terms of the average secrecy rate (ASR) by up to 38.91% when the number of reflecting elements is high (30 elements) compared to a scheme without an IRS. We also observe that the DL-based approach not only provides similar performance to the AO-based scheme but it also significantly reduces computation time.

Consequently, we end up this dissertation by summarizing its main contributions and opening a new door for RL, DL techniques and AI algorithms in future wireless networks.

Contents

List of Figures

List of Tables

Nomenclature

Notation	Description
4G	Fourth-Generation
5G	Fifth-Generation
6G	Sixth-Generation
AC	Actor-Critic
ADC	Analog-to-Digital Converter
AN	Artificial Noise
AO	Alternating Optimization
AoA	Angle of Arrival
AoD	Angle of Departure
API	Application Programming Interface
ASR	Average Secrecy Rate
AWGN	Additive White Gaussian Noise
BP	Back-Propagation
BS	Base Station
CR	Cognitive Radio
CRN	Cognitive Radio Network
CU	Cognitive User
CSCG	Circularly Symmetric Complex Gaussian
CSI	Channel State Information
DDC	Digital Down Converter
DNN	Deep Neural Network
DL	Deep Learning
DSA	Dynamic Spectrum Access
DQN	Deep Q-Network
ED	Energy Detection
EH	Energy Harvesting
ER	Energy Receiver
FFNN	Feed-Forward Neural Network
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
FPP	Feasible Point Pursuit
GMSK	Gaussian Minimum Shift Keying
GNU	GNU's Not Unix
GRC	GNU Radio Companion
HTTP	Hypertext Transfer Protocol
ID	Information Decoding
IFC	Interference Channel
IoT	Internet of Thing
IR	Information Receiver
IRS	Intelligent Reflecting Surface
LoS	Line-of-Sight
LIM	Large Intelligent Metasurface
LIS	Large Intelligent Surface
LRN	Layer Recurrent Network
LSTM	Long Short-Term Memory

MDP	Markov Decision Process
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MSE	Mean Squared Error
NARX	Nonlinear AutoRegressive network with eXogenous inputs
NOMA	Non Orthogonal Multiple Access
OFDM	Orthogonal Frequency Division Multiplexing
OS	Operating System
PU	Primary User
PUE	Primary User Emulation
PS	Power-Splitting
PSD	Positive Semi-Definite
QoS	Quality of Service
QPSK	Quadrature Phase-Shift Keying
REM	Radio Environment Map
ReLU	Rectified Linear Unit
RF	Radio Frequency
RNN	Recurrent Neural Network
RL	Reinforcement Learning
SARSA	State-Action-Reward-State-Action
SCA	Successive Convex Approximation
SE	Spectral Efficiency
SDR	Software-Defined Radio
SNR	Signal-to-Noise Ratio
SISO	Single-Input Single-Output
SINR	Signal-to-Interference-plus-Noise Ratio
SU	Secondary User
SWIG	Simplified Wrapper and Interface Generator
SWIPT	Simultaneous Wireless Information and Power Transfer
TACT	Transfer Actor-Critic
TD	Temporal Difference
TGACT	Transfer Game-Actor-Critic
TS	Time-Switching
UAV	Unmanned Aerial Vehicle
UE	User Equipment
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral
WMMSE	Weighted Minimum Mean Square Error
WCN	Wireless Communication Network
WCS	Wireless Communication System
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Background

1.1.1 Cognitive Radio Network and Simultaneous Wireless Information and Power Transfer

Over the past few years, due to the rapid growth of mobile devices, there has been a dramatic increase in the number of wireless services and applications. Consequently, the demand for spectrum resources has increased, and spectrum scarcity has become a more and more serious problem. To address these emerging issues, researchers have been developing new paradigms in network design. Hence, emerging wireless technologies, such as cognitive radio networks (CRNs) [1,2], were introduced to improve the efficiency in the spatial utilization of the radio spectrum [3]. The basic idea of a CR network is to allow unlicensed radio users, called secondary users (SUs), to share frequencies assigned to licensed users, called the primary users (PUs). In order to avoid interfering with the operations of the licensed user, the SU is allowed to be active when the frequency is not used by the corresponding PU. However, when the presence of the PU is detected, the SU has to vacate the occupied frequency. In addition, in order to access the licensed spectrum, the SUs need to perform the spectrum sensing to identify the available spectrum bands, where temporally no PUs are active.

On the other hand, energy harvesting (EH) is a promising approach to prolong the lifetime of energy constrained wireless networks. Among other renewable energy sources such as wind and solar, background radio-frequency (RF) signals can be a potential new source for wireless power transfer which radiated by ambient transmitters. In addition, RF signals have been widely used as a vehicle for wireless information transmission. As a result, simultaneous wireless information and power transfer (SWIPT) becomes attractive due to its ability to perform wireless power transfer and information transmission simultaneously [4, 5]. It not only brings convenience to mobile users, but also achieves significant benefits in terms of energy consumption, time delay, spectral efficiency, and interference management by superposing power transfer and information.

1.1.2 Machine Learning and deep learning

Machine learning (ML) is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and experience to imitate the way that humans learn, gradually improving its accuracy. ML is an important component of the growing field of data science. Through building a model based on sample data, known as training data, algorithms are trained to make classifications, predictions, or decisions without being explicitly programmed. Moreover, it is able to explore insights among the data due to its ability to process and analyze a numerous amounts of data. In addition, based on different algorithms, the machine can make intelligent decisions through learning from its experiences. ML covers three paradigms which are known as a) supervised learning: where the learning of the model is to be carried out by using input samples and their corresponding outputs, b) unsupervised learning, in which the model learns to distinguish the input samples without any output labels and, c) reinforcement learning (RL), where an agent communicates with an environment and learns to map any input to an action. In this dissertation, we will focus on RL. RL focuses on making suitable decisions that are generated by mapping the situations to actions and evaluating which actions have to be considered for maximizing a long-term reward without prior knowledge of the dynamic of the environment. The techniques for RL are Markov decision process (MDP), Q-Learning, policy learning, actor-critic (AC) and multi-armed bandit (MRB).

In addition, we also focus on deep learning (DL) which can be regarded both as a sophisticated and mathematically complex evolution of ML algorithms. DL is a function of AI that understands the function of human brains and use that understanding to create patterns based on artificial neural networks that contain neurons in multiple layers. Some of the common techniques used for DL are Deep Neural Network (DNN), Recurrent Neural

Network (RNN), Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN).

The main differences between ML and DL are in the following aspects: the ANN algorithm structure, the lower need for human intervention, and the larger data requirements. First and foremost, while traditional ML algorithms have a rather simple structure, such as linear regression or a decision tree, DL is based on an artificial neural network. This multi-layered ANN is, like a human brain, complex and intertwined. Secondly, in DL, the features are extracted automatically, and the algorithm learns from its own errors. Therefore, it requires much less human intervention. Finally, DL requires much more data than a traditional ML algorithm to function properly. ML works with a thousand data points, DL oftentimes only with millions. Due to the complex multi-layer structure, a DL system needs a large dataset to eliminate fluctuations and make high-quality interpretations.

1.2 Thesis Motivation and Objective

In the last few decades, the shortage of radio spectrum resources has become increasingly serious due to the rapid development of multimedia applications and mobile communications. Therefore, network operators are always urged to develop a new network paradigm to effectively implement spectrum allocation. From this perspective, CRN has been emerged as a potential solution that allows SUs to opportunistically access the licensed spectrum when it is temporally unoccupied by PUs [1].

In energy-constrained wireless networks, the user equipments (UEs) are typically powered by batteries that have limited operation time. Although the batteries can be replaced and recharged so that they can prolong the lifetime in the network to a certain extent, it often incurs inconvenient and high costs, e.g., in malicious environments and embedded systems in structure. A safer and more convenient solution is to harvest energy from the surrounding environment, which can provide energy almost perpetually to wireless devices. In addition to non-RF energy sources such as wind and solar, ambient RF signals can be a viable new source for EH. On the other hand, RF signals are also commonly used as a vehicle to transmit information. Therefore, in this case, the RF signals can carry energy and information at the same time. As a result, SWIPT becomes an interesting new area of research that attracts attention. Hence, it is essential to allocate the radio and energy resources efficiently based on AI to improve the lifetime and performance of the network.

The main objective of this dissertation is to solve the aforementioned issues by using AI-based methods, such as the iteration-based dynamic programming, reinforcement learning, and DL. The contributions of this dissertation are summarized as follows:

(i) First, we implement spectrum sensing and share the local spectrum sensing results from the device to the cloud server, and vice versa, using Universal Software Radio Peripheral (USRP) 2900 devices. We also utilize a spectrum sensing system with video transmission on such devices.

(ii) Second, we propose a transfer game-AC learning framework for the security of multi-channel communication in a CRN where various jammers try to access channels of interest to prevent SUs from using them.

(iii) Third, we study the single-input single-output (SISO) SWIPT system with multi-user and aim to optimize transmission power. In addition, the DL-based approach is also proposed to solve the problems of computational power and time.

(iv) Finally, we investigate the physical-layer secure and transmission optimization problem in an IRS-assisted SWIPT system, which aims to maximize the system secrecy rate under energy and power constraints of the UE and transmitter, respectively. The DL is also proposed to improve the computation time.

1.3 Thesis Outline

The rest of this dissertation is organized as follows. Chapter 2 presents testbed implementations of spectrum sensing in CRN using USRP devices. Chapter 3 introduces a transfer games AC learning framework for anti-jamming in multi-channel CRNs. Chapter 4 studies the transmit power optimization problem in a SISO power splitting (PS)-SWIPT system and DL. Chapter 5 investigates a secure-transmission scheme for IRS-assisted SWIPT system and DL. Chapter 6 discusses about the future directions of research. A brief description of each chapter is given below.

Chapter 2 utilizes testbed implementations of spectrum sensing with cloud service and video transmission in CRN. In these implementations, the first one performs collect sensing information and make local sensing decisions at the receiver. Then, the raw sensing information and the temp-sensing decision at the receiver will be uploaded to the ThingSpeak cloud. Meanwhile, the other implements a spectrum sensing system with video transmission in which the video is transmitted and received after spectrum-sensing is implemented at both transceivers. The testbed implementations are developed using USRP and GNU's Not Unix (GNU) Radio as hardware and software platforms, respectively.

Chapter 3 studies the optimal channel selection for the SU in a communication channels CRN under jamming attack. In this network, an SU is transmitting data to a receiver SU while multiple jammers independently perform jamming on legitimate link of transceiver communication. Each jammer attacks a random channel of interest. The security problem is finding the optimal channel to maximize the long-term reward of the SU where communication channels are not used by the PUs and are not jammed by attackers. The problem is formulated and solved by game theory and the transfer game-AC learning framework.

Chapter 4 proposes DL-based approach to fast power allocation in SISO SWIPT systems with a PS scheme. In this network, we first solve the transmit power minimization problem while satisfying sufficient signal-to-interference-plus-noise ratio (SINR) and harvested energy requirements to ensure receiver circuits work in SWIPT systems. Then, we aim to provide a DL solution, which is a promising solution to overcome limitations on computation costs and time of optimization algorithm. DL architectures used in this work include a type of DNN: the Feed-Forward Neural Network (FFNN) and three types of RNN: the Layer Recurrent Network (LRN), the Nonlinear AutoRegressive network with eXogenous inputs (NARX), and LSTM.

Chapter 5 employs an IRS-assisted SWIPT system with PS scheme is installed in the UE. In this network, the system secrecy rate is maximized by jointly finding optimal solutions for transmitter power, PS factor of UE, and phase shifts matrix of IRS under the required minimum energy harvesting and maximum transmitter power. The optimal solutions are obtained by using the alternating optimization (AO)-based scheme. In addition, the DL approach is also proposed to improve the computation time.

Chapter 6 concludes this dissertation and gives a discussion on future research directions.

Chapter 2

Experiments of Spectrum-Sensing, Cloud Uploading and Video Transmission for Cognitive Radio using USRP with GNU Radio

2.1 Introduction

Wireless communications systems have multiplied significantly over the last two decades. However, the radio spectrum used for wireless communications is a finite resource. Therefore, it is necessary to find other systems that can use spectrum efficiently. Currently, technologists are interested in CR because overall efficiency of spectrum utilization can increase significantly. CR enables the usage of temporarily unused frequency bands, commonly known as spectrum holes [6, 7]. Spectrum holes are generally categorized into temporal and spatial spectrum holes. A temporal spectrum hole, which is unoccupied by the PU during certain times, can be used by SUs in the unused time slots. A spatial spectrum hole is a band that is unoccupied by the PU in some geographic areas. Therefore, it can be used by SUs [8–10].

In CR, spectrum sensing is essential, since SUs must determine whether PUs are utilizing the channel or not, and furthermore, SUs opportunistically access the licensed channel when the PU is absent. So far, various approaches have been proposed for spectrum sensing, such as the matched filter, energy detection (ED), feature detection, and more recently, wavelet detection [11]. In these methods, ED uses the energy of the received signal at the SU to determine the presence of signals from the PU [12]. This simple method can gather spectrum occupancy information quickly. It does not require prior information about the behavior (modulation) of the received signal or excessive analog-to-digital converter (ADC) and signal-processing capabilities of the matched filter and cyclostationary detector, respectively; or other features to recognize the primary signal when it appears. Due to the complexity of matched filters, feature detection, and cyclostationary spectrum sensing approaches, those approaches are more precise than ED, but they consume more computational resources in a CRN. Besides, spectrum sensing via ED in CR is included in the IEEE 802.22 standard. Hence, an energy detector is an optimal choice when information about the channel is not available. Table 2.1 lists the schemes used to detect the transmitter signal, along with their pros and cons.

Table 2.1: Primary transmitter detection schemes

Sensing schemes	**Pros**	**Cons**
Energy detection	Easy to implement, no information about the primary signal required	False alarms, uncertainty in noise power
Matched filter	Fewer samples required, less noise effects	Primary signal information is required, synchronization between SUs
Feature	Robustness, fewer false alarms	Cyclostationary signal, consumes computational power

There has been tremendous growth in the fields of multimedia and mobile communications. The convergence of these two fields has resulted in mobile multimedia communications, which has attracted the attention of the research community around the world [13]. A lot of research has been done in this area to find new methodologies to improve or innovate ways to implement the technology with better bandwidth and energy efficiency, because these two resources are limited.

USRP is a hardware platform that allows general-purpose computers to function as high-bandwidth software radios [13]. The application layer communicates with the physical layer through some intermediate layers. For a stationary host, this activity seems to be a good option where the communications protocol is systematic and defined according to the environment where it is located. But for a mobile node, the environment conditions change over time, and hence, the transmit power, bandwidth, and quality of the channel have to be continuously monitored and passed on to the application layer in order to select a suitable algorithm. In turn, the physical layer has to change per the suggestions from the application layer from time to time. GNU Radio and USRP bring the application developer as close to the hardware as to the antenna itself, and provides the user with the flexibility to change the communications parameters on the fly [14].

Besides that, users sometimes do not have the ability to sense the channel, or users may need sensing information for performance analysis. To do this, the sensing information should be saved where everyone can easily access spectrum sensing results to further utilize them. The cloud server is the solution to this problem. A cloud server allows users to store data online, which they can access from any location via the internet. So, the sensed information can be shared with other users. From that, users can use the sensed information that is uploaded to the cloud server to estimate the statistics of the primary network, such as the probability of absence and the transition probability for the PU. Then, users can analyze and improve the performance of local sensing. To successfully integrate CRs into the cloud computing paradigm, we also implement cloud computing–based spectrum sensing by utilizing a cloud server with ThingSpeak, such that we can store, process, and share the sensed information more efficiently in a centralized way. In addition, due to the dynamics of the primary network, if the time for delivering sensing information to the cloud server involves a long delay, the immediate sensed information/global decision may be meaningless. Therefore, in order to utilize the sensed information/global decisions that are shared in the cloud, the delay time should be very short in comparison with the whole time slot. Because of the limit in the time delay until sensing information is delivered to the cloud server, in this chapter, we mainly focus on ways to implement local spectrum sensing on the user side to upload to, and share the local spectrum sensing results from, the cloud server and to estimate the performance of local sensing based on absence and transition probabilities.

The contributions of this chapter are summarized as follows:

- We implement a CR system by using USRP National Instruments (NI) 2900 devices, and we evaluate the sensing performance of ED such as the probability of detection, and the probability of false alarm at USRP receiver in a real environment.

- We also focus on ways to implement local spectrum sensing on the user side to upload and share the local spectrum sensing results to the cloud server, and vice versa to estimate the performance of spectrum sensing.
- In addition, to utilize mobile multimedia communications, a spectrum sensing system based on ED combined with video transmission is developed using USRP and GNU Radio as hardware and software platforms, respectively. The video is transmitted and received after spectrum sensing is implemented at both the transmitter and receiver.

This chapter is organized as follows. Section 2.2 presents related works and background. In Section 2.3, we describe system design and experiments. In Section 2.4, we evaluate experiment results and discussions. Finally, we provide a conclusion in Section 2.5.

2.2 Related Works and Background

In CR, spectrum-sensing based on ED has been proposed and widely used because it does not require transmitted signal properties, channel information, or even the type of modulation. Abdulsattar and Hussein [6] presented a survey of energy detectors over additive white Gaussian noise (AWGN) on different fading channels for spectrum-sensing methodologies in CR. Theoretical analysis of time domain ED and threshold setting was investigated.

Ma *et al.* [8] described the fundamental signal-processing aspects involved in developing a fully functional CR network, including basic techniques of spectrum-sensing such as energy, cyclostationary, pilot-based coherent, covariance-based, and wavelet-based detection, that can effectively improve the overall detection capability of a CR network. Besides that, spectrum sculpting was also discussed for avoiding interference with the PUs.

Cabric *et al.* [11] explored the new field of CR with a special emphasis on one unique aspect: spectrum sensing. Three digital signal processing techniques (matched filtering, ED, and cyclostationary feature detection) were investigated, which identified two key issues related to the CR front end: dynamic range reduction and wideband frequency agility.

The USRP platform is a low-cost and high-quality realization of SDR. It allows users various functionalities to achieve efficient, real-time realization of very complicated wireless systems that operate in the RF band. The USRP platform converts the digital baseband signal delivered from the computer to an analog signal in the RF band. USRP includes two main boards: the daughterboard and the motherboard [13]. A programmable USB 2.0 controller communicates between USRP and GNU Radio and a field programmable gate array (FPGA) for implementing four digital down converters (DDCs) and high-rate signal processing. In our experiments, two USRP boards were used: the PU signal was a generated signal for the first board, whereas the second was used for spectrum-sensing purposes, and acted as the SU. The USRP NI 2900 device is an RF transceiver capable of transmitting and receiving signals from 70 MHz to 6 GHz, which means that many applications can be available by utilizing the frequency band from FM radio through to Wi-Fi bands. The USRP NI 2900 has two connectors: RX2 (input terminal for the RF signal) and TX1 RX1 (input and output terminals for the RF signal). These connectors go to an RF switch on board, which allows transmit and receive operations to occur on the same shared antenna.

Unlike other off-the-shelf options, the USRP family provides a complete and versatile solution with software that accelerates development. All software processing was realized in the open-source GNU Radio environment [15]. These libraries, together with the appropriate drivers for manipulating the USRP boards and graphical programming environment, allow efficient and accurate implementation of the selected spectrum-sensing algorithms. All signal processing was realized in the GNU Radio environment with the graphical tool called GNU Radio Companion (GRC), where the whole system is built from blocks. Besides that, applications can be created using the Python script language behind the blocks. The performance-critical signal processing path is implemented in C++. A simplified wrapper and interface generator (SWIG) interface, which is an interface compiler, is used to link C++ with Python.

Sowmiya and Sangeetha [16] analyzed a CR system based on ED with modulation by using USRP NI 2920 devices. In that experiment, the quadrature phase-shift keying (QPSK) modulated signal was received and the energy analyzed to detect the presence of PUs.

Sensing performance in a real-time testbed of a GNU Radio and USRP software-defined radio (SDR) communications platform was implemented [17]. Rashid *et al.* investigated two main performance metrics for spectrum-sensing: probability of false alarm (P_{fa}) and probability of detection, P_d. P_{fa} is used to determine the threshold of the CR dynamic spectrum access (DSA) system, while P_d is used to determine how many samples are needed by CR DSA to meet the desired performance.

The ThingSpeak application programming interface (API) is an open source interface that listens to incoming data, timestamps it, and outputs it for both human users (through visual graphs) and machines (through easily parseable code) [18, 19]. The cloud brings everything together and allows us to interact with things and, even more interestingly, allows things to interact with other things. For connection to the cloud server, we focus on the ThingSpeak API. The interface for ThingSpeak capabilities communicates with objects, as well as interesting additional applications. Moreover, ThingSpeak allows us to collect data using sensors and display them in graphs. It offers near–real-time data collection, data processing, and simple visualizations for users. Data are stored in so-called channels, which provide the user with a list of features [19]. Each channel can store up to eight fields of data, using up to 255 alphanumeric characters each. All incoming data are time- and date-stamped, and receive sequential IDs. When a channel has been created, data can be published by accessing the ThingSpeak API with a *write key.* Consequently, a *read key* is used to access channel data when it is set to keep the data private (the default setting). Channels can change to public status, in which case no read key is required. The first steps to setting up ThingSpeak are always the same. After signing up for a new user account, we can log in and create new channels [19]. After that, we can create a new channel by selecting Channels, then choosing My Channels and Create New Channel. The channel has a unique identifier key that is used to identify the channel when reading or uploading data.

According to the ThingSpeak website, the API works as noted in Fig. 2.1 [19]. *Things* are objects, such as data, that are collected from sensors. Data are sent and received via simple Hypertext Transfer Protocol (HTTP) POSTs, much like going to a web page and filling out a form. This communication is implemented through plain text, JavaScript Object Notation (JSON) or Extensible Markup Language (XML). The data are then uploaded to the cloud and, from there, can be used for a variety of purposes.

2.3 System Design and Experiments

In this section, we first present the local spectrum sensing. Secondly, we implement cloud-based spectrum sensing where a USRP NI 2900 receiver with GNU is used to collect sensing information and make local sensing decisions for local/immediate usage. The raw

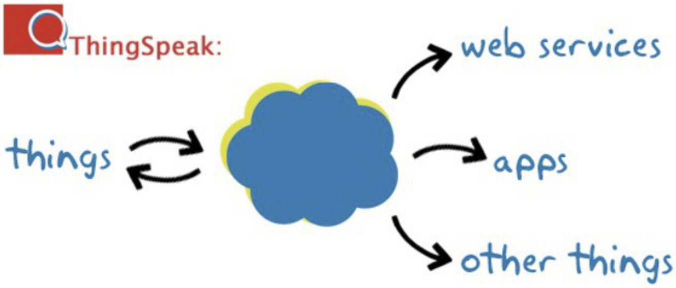


Figure 2.1: ThingSpeak with a cloud interface.

sensing information and the temp-sensing decision at the URSP receiver will be uploaded to the ThingSpeak cloud. After that, the cloud will process the received information to make a global decision in order to share it with in-demand users on the network. Finally, we implement a spectrum sensing system based on ED combined with video transmission in which the video is transmitted and received after spectrum-sensing is implemented at both the transmitter and receiver.

2.3.1 Local Spectrum Sensing

Energy detection is used for spectrum sensing in the USRP device, in which two hypotheses about the PU signal (absence (H_0) and presence (H_1)) are considered [16]. The received signal, $x\left(n\right)$, can be expressed as:

$$x\left(n\right)=\begin{cases} w\left(n\right) & H_0, \\ hs\left(n\right)+w\left(n\right) & H_1, \end{cases} \tag{2.1}$$

where $n=1,\ldots,N$; N is the number of samples; h is the amplitude gain of the channel, which is assumed to be 0 and 1 under hypothesis H_0 and H_1, respectively; $w\left(n\right)$ is AWGN with zero mean and variance σ_w^2; and $s\left(n\right)$ is the transmitted signal. The principle of the spectrum sensing operation is to decide between H_0 and H_1 based on observation of received signal $x\left(n\right)$.

The presence of the PU can be detected simply by calculating the amount of received power in the considered frequency band and comparing it with a set threshold. The algorithm for ED will make a decision on spectrum occupancy by the PU if the received signal power is greater than the threshold. The mean power of N samples is collected and can be represented as random variable:

$$P_N = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2, \tag{2.2}$$

where P_N is the mean output of the energy detector. In traditional ED, the reliability of the spectrum sensing algorithm is crucially influenced by the threshold. Then, based on (2.2), generic decision rule D_N can then be modified to the considered case:

$$D_N = \begin{cases} H_0, & \text{if } P_N \leq \gamma, \\ H_1, & \text{otherwise } P_N > \gamma. \end{cases} \tag{2.3}$$

The performance of the energy detector is characterized by two metrics: probability of detection, P_d, which occurs when decision D_N is H_1, and the PU signal is practically present (H_1); and probability of false alarm, P_{fa}, which corresponds to the decision H_1 but the PU signal is absent (H_0) [16]:

$$P_{fa} = P\left(P_N > \gamma | H_0\right), \tag{2.4}$$

$$P_d = P\left(P_N > \gamma | H_1\right). \tag{2.5}$$

If the noise term is assumed to be a circularly symmetric complex Gaussian (CSCG), using Gaussian distribution approximation for the probability density function of P_N, it can be derived from (2.4) and (2.5) [20]:

$$P_{fa} = Q\left(\frac{\gamma - 2N\sigma_w^2}{\sqrt{4N\sigma_w^4}}\right), \tag{2.6}$$

$$P_d = Q\left(\frac{\gamma - 2N\left(SNR+1\right)}{\sqrt{4N\left(2SNR+1\right)}}\right), \tag{2.7}$$

where SNR and $Q(\cdot)$ represent signal-to-noise ratio and the Q-function, respectively. The challenge in local spectrum-sensing is to reliably decide on the two hypotheses to achieve high P_d for good protection of the PU, and low P_{fa} to provide satisfactory access for SUs.

2.3.2 Spectrum Sensing Implementation and ThingSpeak Uploading

In this section, two USRP boards were used: the PU signal was a generated signal for the first board, whereas the second was used for spectrum-sensing purposes, and acted as the SU, was also used to upload data to ThingSpeak. The schematic diagram and dedicated photographs of the experiment are shown in Fig. 2.2 and Fig. 2.3, respectively. The experimental system consists of two USRP devices, one laptop, and one PC.

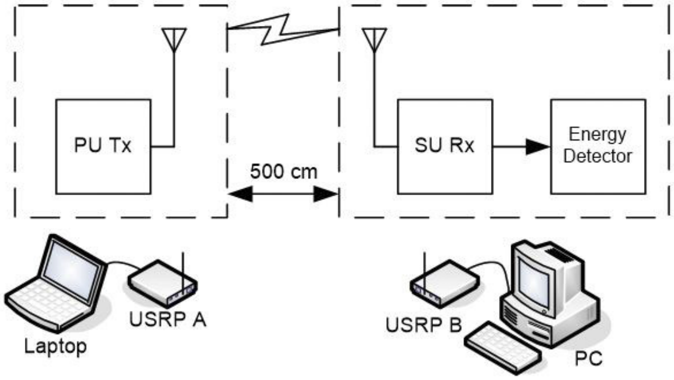


Figure 2.2: Schematic diagram of the system.

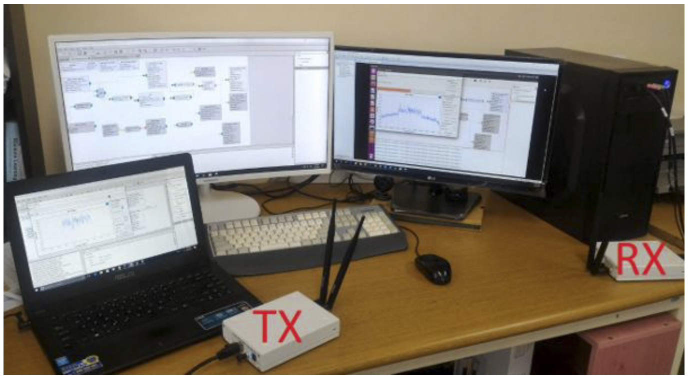
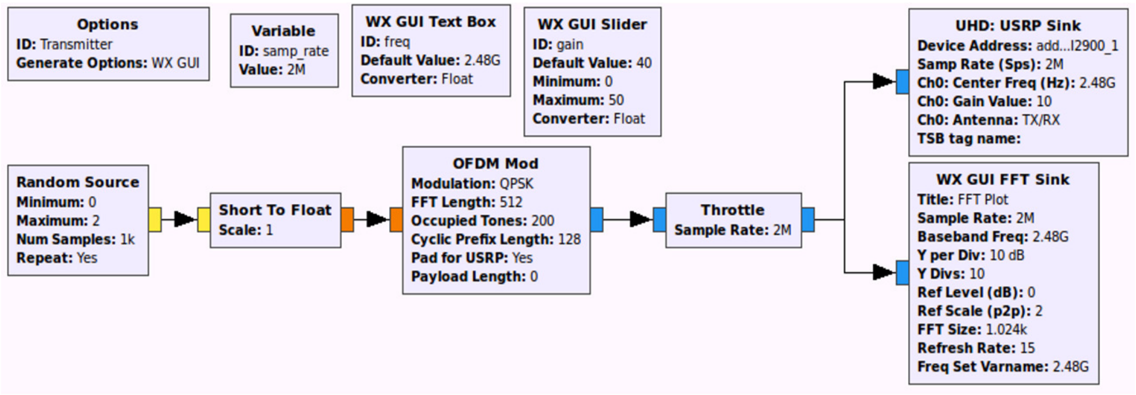


Figure 2.3: PU transmitter and SU receiver with USRP hardware and GNU Radio software.

The laptop with USRP A acts as the transmitter while the PC with USRP B acts as the receiver. There is interference at neighboring access points, which can affect the USRP frequencies. Therefore, the USRP center frequency was set at 2.48 GHz to avoid interference and jamming with the said access point operating in the 2.4 GHz band. The energy detector was implemented using 1024-point fast Fourier transform (FFT). Each block of FFT output was averaged and stored inside a buffer. N averaged FFT blocks were collected and then averaged again to acquire the final result used to make the decision on the presence of the PU.

On the transmitter side, the spectrum of the multicarrier signal was tested in



Options
ID: Transmitter
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 2M

WX GUI Text Box
ID: freq
Default Value: 2.48G
Converter: Float

WX GUI Slider
ID: gain
Default Value: 40
Minimum: 0
Maximum: 50
Converter: Float

UHD: USRP Sink
Device Address: add...I2900_1
Samp Rate (Sps): 2M
Ch0: Center Freq (Hz): 2.48G
Ch0: Gain Value: 10
Ch0: Antenna: TX/RX
TSB tag name:

Random Source
Minimum: 0
Maximum: 2
Num Samples: 1k
Repeat: Yes

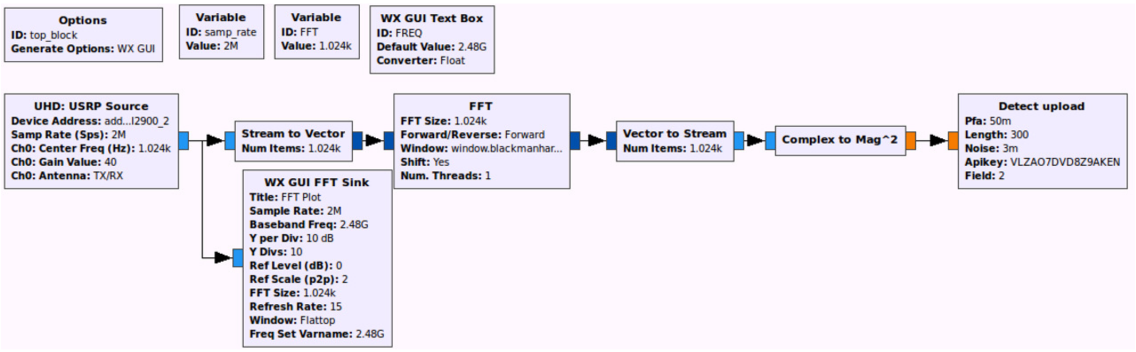
Short To Float
Scale: 1

OFDM Mod
Modulation: QPSK
FFT Length: 512
Occupied Tones: 200
Cyclic Prefix Length: 128
Pad for USRP: Yes
Payload Length: 0

Throttle
Sample Rate: 2M

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 2M
Baseband Freq: 2.48G
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: 2.48G

Figure 2.4: The PU transmitter with OFDM modulation in the GNU Radio Companion (GRC).



Options
ID: top_block
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 2M

Variable
ID: FFT
Value: 1.024k

WX GUI Text Box
ID: FREQ
Default Value: 2.48G
Converter: Float

UHD: USRP Source
Device Address: add...I2900_2
Samp Rate (Sps): 2M
Ch0: Center Freq (Hz): 1.024k
Ch0: Gain Value: 40
Ch0: Antenna: TX/RX

Stream to Vector
Num Items: 1.024k

FFT
FFT Size: 1.024k
Forward/Reverse: Forward
Window: window.blackmanhar...
Shift: Yes
Num. Threads: 1

Vector to Stream
Num Items: 1.024k

Complex to Mag^2

Detect upload
Pfa: 50m
Length: 300
Noise: 3m
Apikey: VLZAO7DVD8Z9AKEN
Field: 2

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 2M
Baseband Freq: 2.48G
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Window: Flattop
Freq Set Varname: 2.48G

Figure 2.5: The SU receiver with energy detection and data uploading function block in the GRC.

this research work–the orthogonal frequency division multiplexing (OFDM) symbol with $N_{OFDM} = 512$ subcarriers. The PU transmitter side with OFDM modulation in the GRC is shown in Fig. 2.4. The signal source block (*Random Source*) generated repeatedly random data, which were mapped to QPSK symbols and were then subject to the OFDM modulation block (realized in *OFDM Mod*). Finally, after power was adjusted properly, the signal was sent to the local spectrum analyzer block (*FFT Sink*) and to the USRP block (*USRP Sink*) responsible for sending data. Notice that the complex sampling frequency is set to 2 MHz, the antenna gain is set to 40 dB, and the center frequency is set to 2.48 GHz. Analogous to the PU transmitter, the SU receiver was also realized in the GNU Radio environment. A schematic diagram of the SU receiver is shown in Fig. 2.5. The parameters at the receiver were also set to the same center frequency, sample rate, gain, and FFT size. In addition, the receiver included parameters like the given P_{fa} ($P_{fa} = 0.05$) and the number of sensing samples ($N = 300$) for local spectrum sensing. Data from the RF spectrum were collected by the *USRP Source* block. It operates at the center frequency equal to 2.48 GHz. Afterwards, the signal goes through chains with ED. We can observe the received signal through the *FFT Sink* block.

Next, we focus on the ED algorithm. The signal is transformed to the frequency domain by the FFT block. After that, these data can be analyzed and stored in the *Detect upload* block, which has two main functions: (i) processing and storing data into a data file, and then, performance of the ED algorithm is estimated by using Matlab; and (ii) providing a connection between the GNU system and ThingSpeak to upload P_d to ThingSpeak.

The *Detect upload* block not only implements the ED algorithm but also connects the ED system and ThingSpeak. This block will use a *write API key* ("VLZAO7DVD8Z9AKEN") to connect to ThingSpeak. All the collected data are stored in text files, which are integrated into the *Detect upload* block and will be uploaded to ThingSpeak. Transmitter and receiver parameters are shown in Table 2.2. The *Detect upload* block is shown in Fig. 2.6.

Table 2.2: Parameters for the transmitter and receiver

Parameter	**Transmitter**	**Receiver**
Center frequency (GHz)	2.48	2.48
Sample rate (MHz)	2	2
Gain (dB)	40	40
FFT	1024	1024
Number of sensing samples		300
Given propability of false alarm		0.05
Noise (dB)		0.003
Write API key		VLZA07DVD8Z9AKEN

2.3.3 Spectrum Sensing Implementation and Video Transmission

A schematic diagram and dedicated photograph of spectrum sensing with video transmission and reception which consisted of two USRPs, one laptop, and one PC are shown in Fig. 2.7 and Fig. 2.8, respectively. Note that ED in this case is already built into

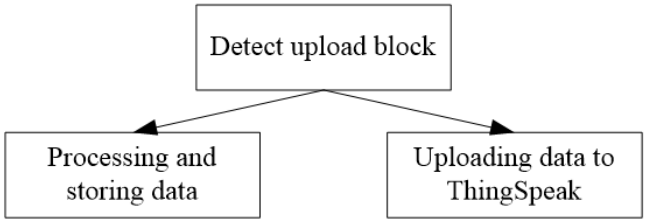


Figure 2.6: The detect upload function block.

the python script file. Therefore, the schematic system diagram in Fig. 2.7 has no energy detector block as compared with Fig. 2.2. The whole system of spectrum sensing with video transmission and reception is shown in Fig. 2.9.

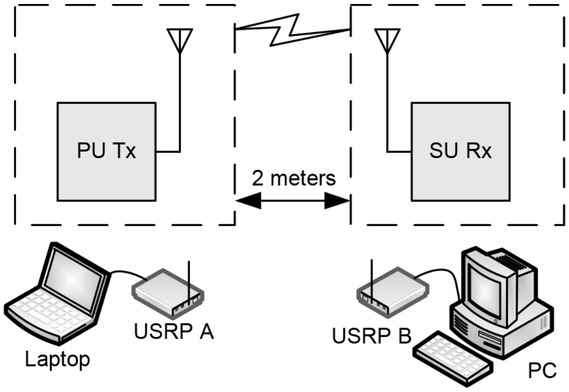


Figure 2.7: Schematic system diagram.

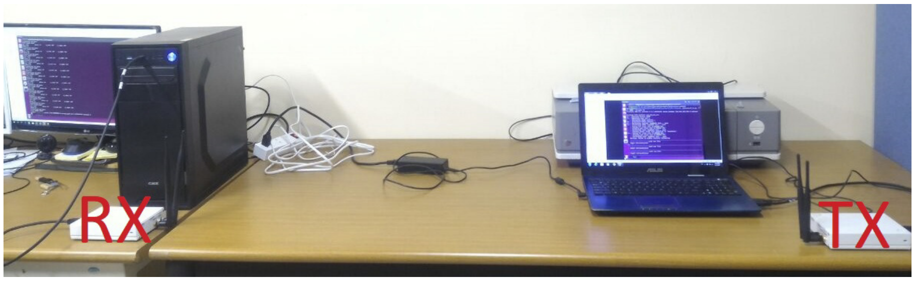


Figure 2.8: Spectrum sensing, video transmission and reception with USRP hardware and GNU Radio software.

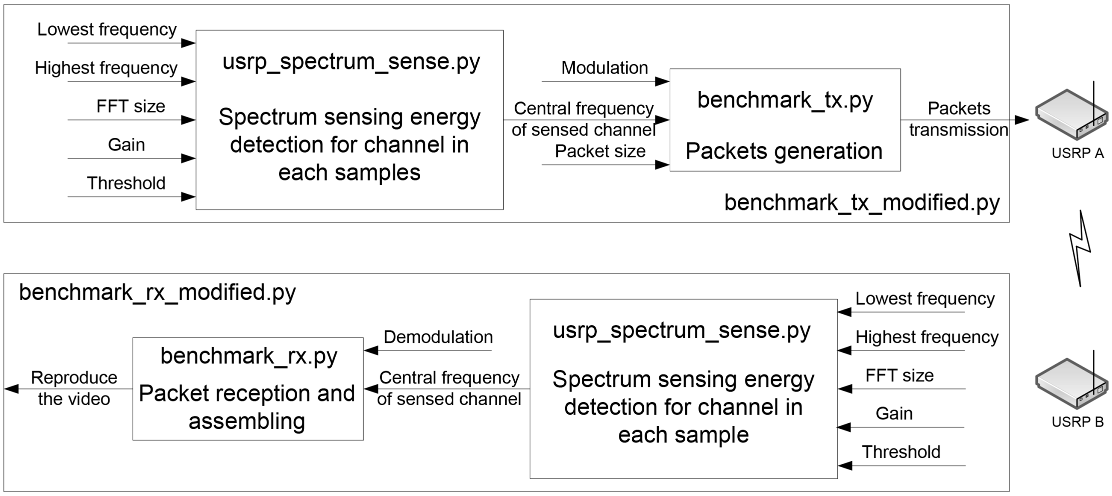


Figure 2.9: The whole system of spectrum sensing with video transmission and reception.

2.3.3.1 Spectrum Sensing Implementation

The script *usrp_spectrum_sense.py* was considered for the design of the spectrum-sensing functionality implemented in the testbed; it can be found in the toolkit provided by the GNU Radio software. This script has been used as basic code for implementing a wideband spectrum analyzer to properly sense the desired spectrum bands. However, it was extended to properly sense the spectrum bands considered in the different spectrum management strategies implemented in the testbed. The script receives different input parameters from the user, such as the lowest and highest frequencies of the band to be sensed; the FFT size parameter for the number of samples; the bandwidths considered in performing the magnitude analysis of the sensed signal; gain; and threshold. Then, the script computes the signal energy detected in each sample during execution of the spectrum sensing functionality. Finally, the output of the script provides the center frequency at the sensed channel.

Both transmitter and receiver also implemented spectrum sensing for the sensed channel that is free or detected to transmit or receive video. The transmitter will sense the channel, and if the channel is free (its mean average signal is less than a given threshold based on measurements), a video signal is transmitted on the sensed frequency. Conversely, if the channel is busy, a video signal will be transmitted on the next frequency [21]. The USRP receiver also senses the channel, but in contrast, if the channel is detected, the signal is demodulated to reproduce the transmitted video signal. Fig. 2.10 shows the flow chart for sensing the channel in the system.

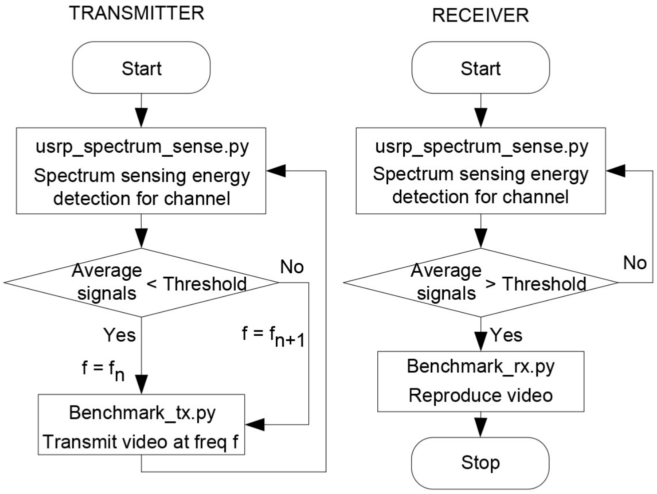


Figure 2.10: The flow charts of video transmission and reception that are integrated for sensing the channel in the system.

2.3.3.2 Video Transmission

The data transmission and receiving functionalities were implemented through GNU Radio's *benchmark_tx.py* and *benchmark_rx.py* scripts, respectively. They are also found in the toolkit provided by the GNU Radio software. In this section, we integrated *usrp_spectrum_sense.py*, *benchmark_tx.py*, and *benchmark_rx.py* scripts into *benchmark_tx_modified.py* and *benchmark_rx_modified.py* scripts, so implementation of spectrum sensing with video transmission and receiving is more convenient. These scripts take the following input parameters from users: lowest and highest frequency, FFT size, transmitter and receiver gain, and threshold for the spectrum-sensing functionality; a modulation scheme for Gaussian minimum shift keying (GMSK); the sensed frequency from the spectrum sensing; and packet size (only for the script *benchmark_tx.py*).

On the transmitter side, after implementation of spectrum sensing, tasks of the *benchmark_tx* Python file are as follows.

- Data files are read frame by frame from hard disk, and then each frame is packed with packet sequence numbers as the application layer header.
- The application layer packets are sent to the data link layer and the physical layer,

where information on the preamble, access codes, cyclical redundancy checking, etc. is assembled.

- In the last step, the assembled packets are modulated with GMSK and sent to the USRP through a USB interface. USRP then transmits each packet on the sensed channel frequency.

On the receiver side, after the signal is processed with spectrum sensing, the *benchmark_rx* Python file will be implemented. Tasks of this script are as follows.

- Read data from the USRP through the USB interface, which are demodulated with GMSK
- Then, the received link layer packets are disassembled into application layer packets with packet sequence numbers.
- In the last step on the receiver side, all correctly received packets are saved to a file, displaying the decoded video sequences in real time on a display terminal.

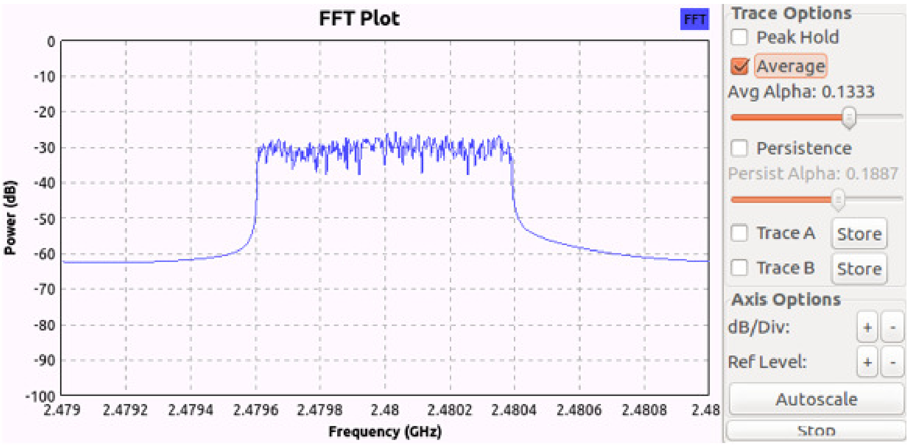
In this research, we set statements at the transmitter as *./benchmark_tx_modified.py 893M 911M --fft 512 -g 60 -q 10* and at the receiver as *./benchmark_rx_modified.py 893M 911M --fft 512 -g 30 -q 4*, where 893M and 911M are the lowest and highest frequency in megahertz (MHz), *fft* is FFT size, g is gain in decibels (dB), and q is the threshold in dB. All parameters can change for each requirement from users.

2.4 Experiments Results and Discussions

2.4.1 Experiments on Spectrum Sensing Implementation and ThingSpeak Uploading

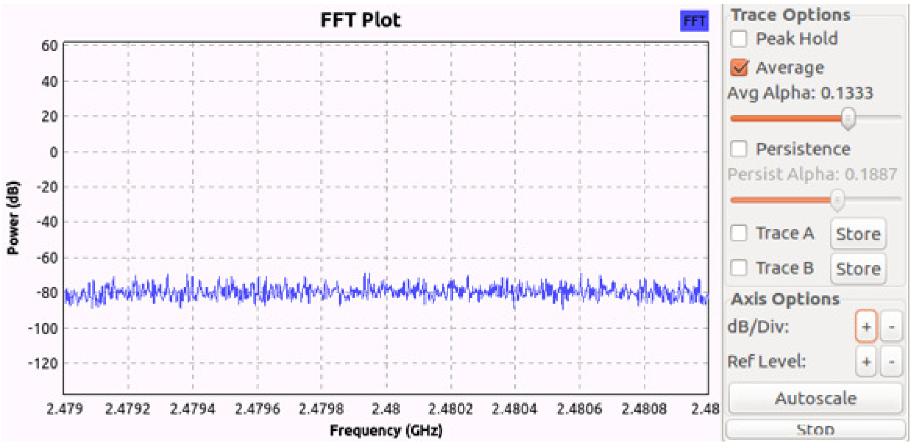
We tested a QPSK signal centered on the 2.48 GHz carrier frequency. From the experiment, power transmission is shown in Fig. 2.11. On the receiver side, the sensing information, when there is no signal transmission from the PU, is shown in Fig. 2.12. Fig. 2.13 shows the SU signal on the receiver side when the PU transmitter utilizes the channel.

In GNU Radio, we can observe received signals based on the FFT sink block. Moreover, the processed signal at the SU is also stored in a data file that is integrated into the *Detect upload* block, and then, we used this data file for analysis in Matlab. Fig.



Trace Options
☐ Peak Hold
☑ Average
Avg Alpha: 0.1333
☐ Persistence
Persist Alpha: 0.1887
☐ Trace A
☐ Trace B
Axis Options
dB/Div:
Ref Level:

Figure 2.11: Transmitted power at the PU transmitter.

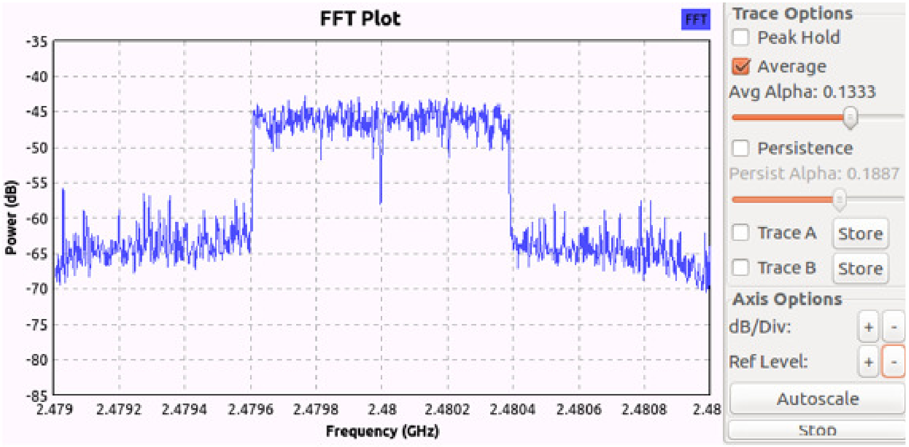


Trace Options
☐ Peak Hold
☑ Average
Avg Alpha: 0.1333
☐ Persistence
Persist Alpha: 0.1887
☐ Trace A
☐ Trace B
Axis Options
dB/Div:
Ref Level:

Figure 2.12: The received signal power at center frequency $f_c = 2.48$ GHz when the signal of the PU is absent.

2.14 shows the probability of detection according to the number of sensing samples. It shows that if we use more samples, the performance of the energy detector is better. Fig. 2.14 also shows that practical and theoretical results are almost similar. Fig. 2.15 shows the probabilities of detection and false alarm according to SNR. When SNR is higher, the performance in probability of detection and probability of false alarm improves more.

In the next step, using ThingSpeak, we integrated data uploading assigned to the *Detect upload* block with ED. As a result of the experiment, the probability of detection versus the number of sensing samples was uploaded to ThingSpeak for saving and sharing via the cloud, as shown in Fig. 2.16. Data that can be downloaded from the cloud by the



Trace Options

☐ Peak Hold

☑ Average

Avg Alpha: 0.1333

☐ Persistence

Persist Alpha: 0.1887

☐ Trace A

☐ Trace B

Axis Options

dB/Div:

Ref Level:

Figure 2.13: The received signal power at center frequency $f_c = 2.48$ GHz when the signal of the PU is present.

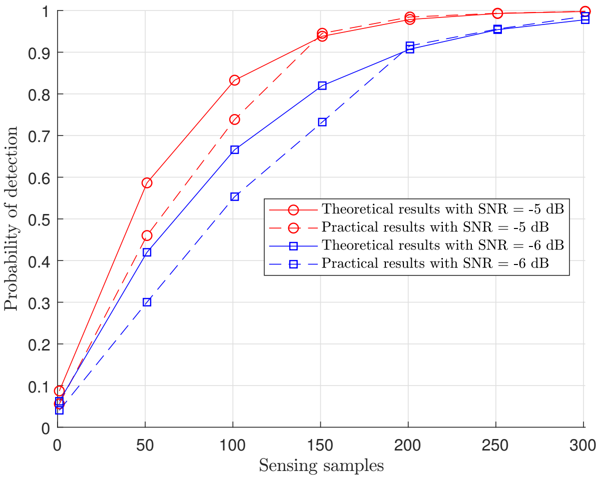


Figure 2.14: The probability of detection according to the number of sensing samples when $P_{fa} = 0.05$.

end user are shown in Fig. 2.17.

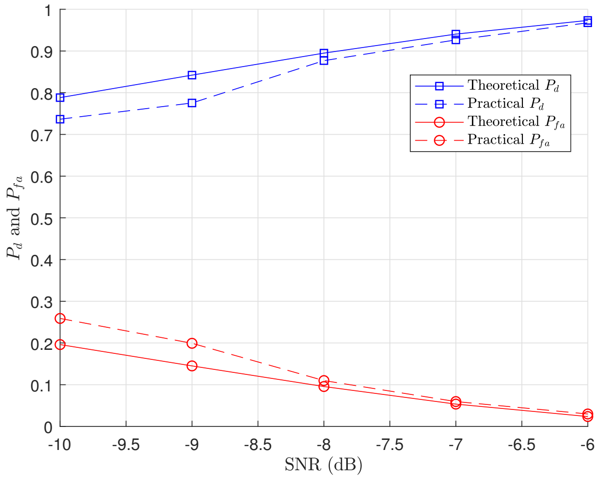


Figure 2.15: The probabilities of detection and false alarm based on SNR.

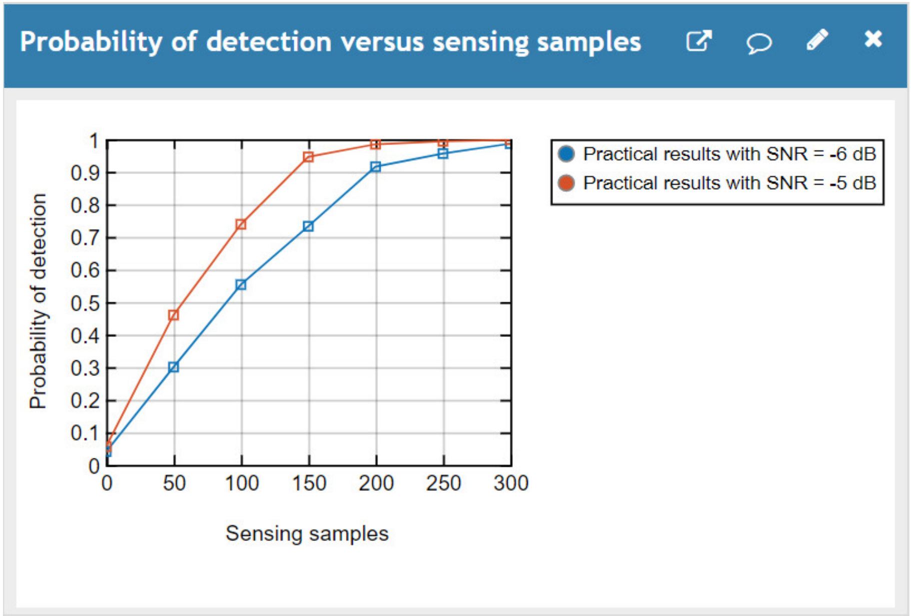

Probability of detection versus sensing samples
Practical results with SNR = -6 dB
Practical results with SNR = -5 dB
Probability of detection
Sensing samples
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
50
100
150
200
250
300

Figure 2.16: The probability of detection uploaded via ThingSpeak to the cloud. From top to bottom: the first line is for SNR = -5 dB and the second line is for SNR = -6 dB.

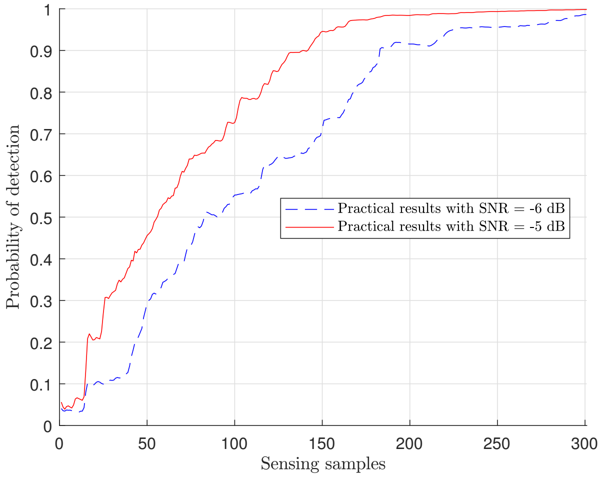


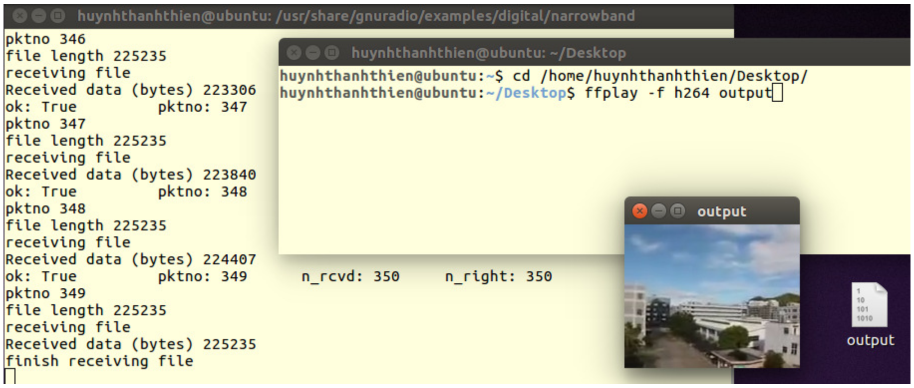
Figure 2.17: The spectrum sensing data received from the cloud by the end user.

2.4.2 Experiments on Spectrum Sensing Implementation and Video Transmission

2.4.2.1 Implementation of Video Transmission with H.264 and MP4 on a Specific Channel

In this testbed, we used two computers, one PC (the receiver) had the Linux operating (OS) system installed, and one laptop (the transmitter), also had the Linux OS. For more conveniently displayed video, we used a ffplay platform to show the video. In this scheme, we transmitted under two scenarios: first, a video in H.264 format was 225,235 bytes in size, and second was a video in MP4 format in a higher resolution at 9.3 MB in size, setting statements as *./benchmark_tx.py –f 900M* and *./benchmark_rx.py –f 900M* for the transmitter and the receiver, respectively. The received videos in H.264 and MP4 formats with information about received packets, file length, and status of reception are shown in Fig. 2.18 and Fig. 2.19, respectively.

For evaluation of performance in the testbed, we considered received packet and right received packet parameters. At the transmitter, we assembled the 225,235 bytes into 350 packets, and we then estimated performance of the testbed based on the number of received packets and right received packets. We estimated performance of the testbed under two scenarios: first, the distance between transmitter and receiver was fixed at 2 meters, with increasing transmitter gain; second, transmitter gain was fixed, but with changes in

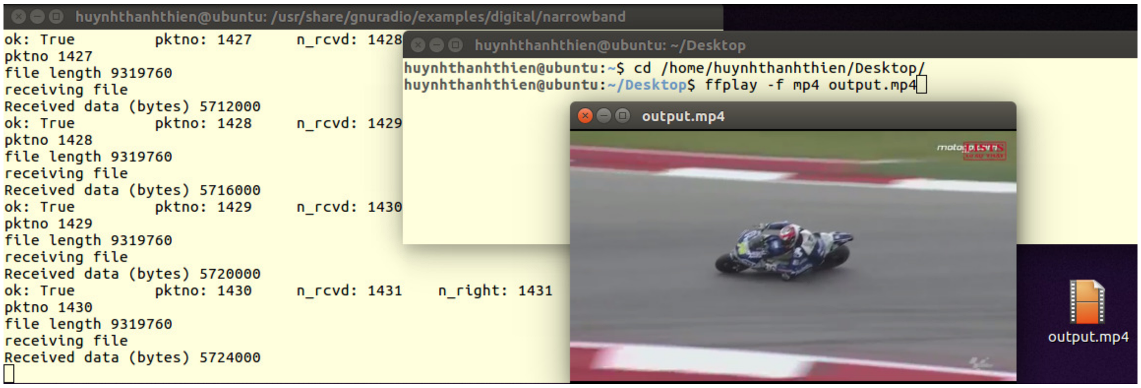


huynhthanhthien@ubuntu: /usr/share/gnuradio/examples/digital/narrowband
huynhthanhthien@ubuntu: ~/Desktop
output
output

```
pktno 346
file length 225235
receiving file
Received data (bytes) 223306
ok: True          pktno: 347
pktno 347
file length 225235
receiving file
Received data (bytes) 223840
ok: True          pktno: 348
pktno 348
file length 225235
receiving file
Received data (bytes) 224407
ok: True          pktno: 349      n_rcvd: 350     n_right: 350
pktno 349
file length 225235
receiving file
Received data (bytes) 225235
finish receiving file
```

```
huynhthanhthien@ubuntu:~$ cd /home/huynhthanhthien/Desktop/
huynhthanhthien@ubuntu:~/Desktop$ ffplay -f h264 output
```

Figure 2.18: Video received in real time in H.264 format on the specific channel (frequency = 900 MHz).



huynhthanhthien@ubuntu: /usr/share/gnuradio/examples/digital/narrowband
huynhthanhthien@ubuntu: ~/Desktop
output.mp4
output.mp4

```
ok: True          pktno: 1427     n_rcvd: 1428
pktno 1427
file length 9319760
receiving file
Received data (bytes) 5712000
ok: True          pktno: 1428     n_rcvd: 1429
pktno 1428
file length 9319760
receiving file
Received data (bytes) 5716000
ok: True          pktno: 1429     n_rcvd: 1430
pktno 1429
file length 9319760
receiving file
Received data (bytes) 5720000
ok: True          pktno: 1430     n_rcvd: 1431    n_right: 1431
pktno 1430
file length 9319760
receiving file
Received data (bytes) 5724000
```

```
huynhthanhthien@ubuntu:~$ cd /home/huynhthanhthien/Desktop/
huynhthanhthien@ubuntu:~/Desktop$ ffplay -f mp4 output.mp4
```

Figure 2.19: Video received in real time at a higher resolution and in MP4 format on the specific channel (frequency = 900 MHz).

transmit distance. In both cases, receiver gain was fixed at 30 dB. The results of the two scenarios are shown in Fig. 2.20 and 2.21, respectively. Fig. 2.20 and 2.21 show number of transmitted packets (left side column), received packets (middle column), and right received packets (right side column) in the same considered value. Due to influence of some factors such as transmission distance, receiver and transmission gains, some packets will be lost in transmission process. In case receiver can not decode packets, number of right received packets will be less than number of received packets. Fig. 2.20 shows that the more transmitter gain we use, the lower the packet error rate. Besides that, packet error rate can also be estimated based on the transmission distance between transmitter and receiver: the farther the distance, the higher the packet error rate as shown in Fig. 2.21. Note that, packet error rate can be calculated based on the number of received packets compared to the number of transmitted packets.

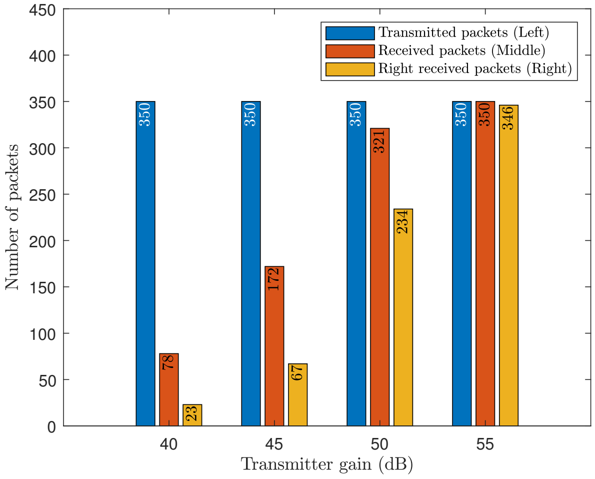


Figure 2.20: Number of received packets, right received packets, and transmitted packets with changes in transmitter gain but a fixed receiver gain of 30 dB.

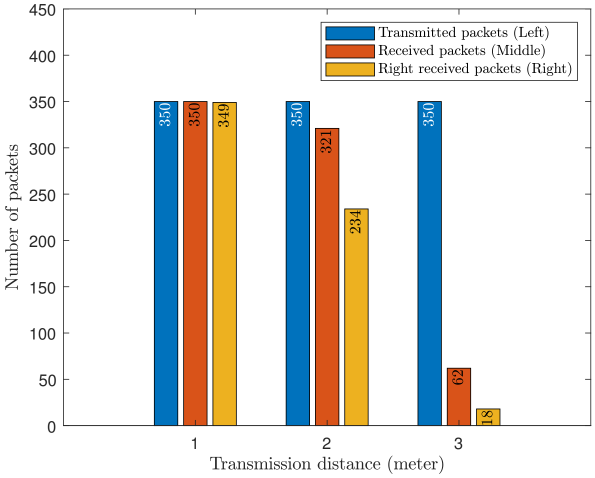
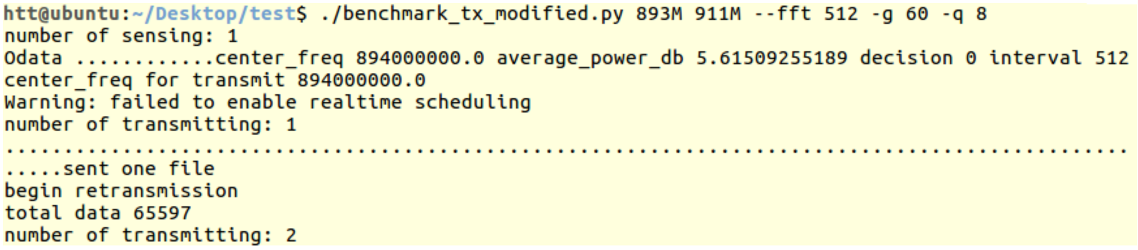


Figure 2.21: Number of received packets, right received packets, and transmitted packets with changes in transmission distance, but with fixed receiver and transmitter gain (30 dB and 50 dB, respectively).

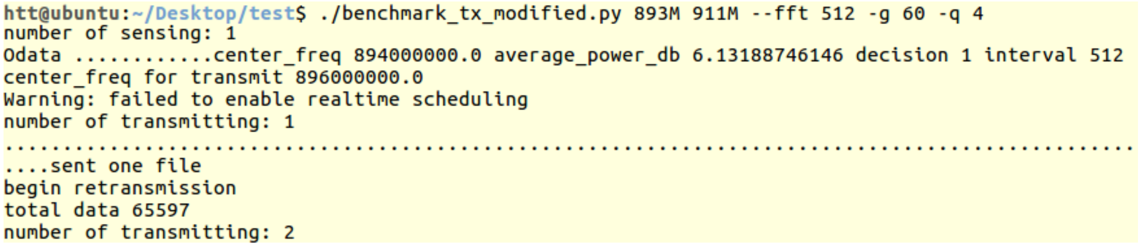
2.4.2.2 Implementation of Spectrum Sensing with Video Transmission at Transmitter and Receiver

We set up the system with input parameters for transmitter and receiver as explained in the video transmission part of Section 2.3.3.2. The input of transmitter Python script and the results of spectrum sensing on the transmitter side at thresholds equal to 8 dB and 4 dB are shown in Fig. 2.22 and Fig. 2.23, respectively.



```
htt@ubuntu:~/Desktop/test$ ./benchmark_tx_modified.py 893M 911M --fft 512 -g 60 -q 8
number of sensing: 1
Odata ...........center_freq 894000000.0 average_power_db 5.61509255189 decision 0 interval 512
center_freq for transmit 894000000.0
Warning: failed to enable realtime scheduling
number of transmitting: 1
................................................................................................
.....sent one file
begin retransmission
total data 65597
number of transmitting: 2
```

Figure 2.22: Spectrum sensing and video transmission at the transmitter with a threshold of 8 dB.

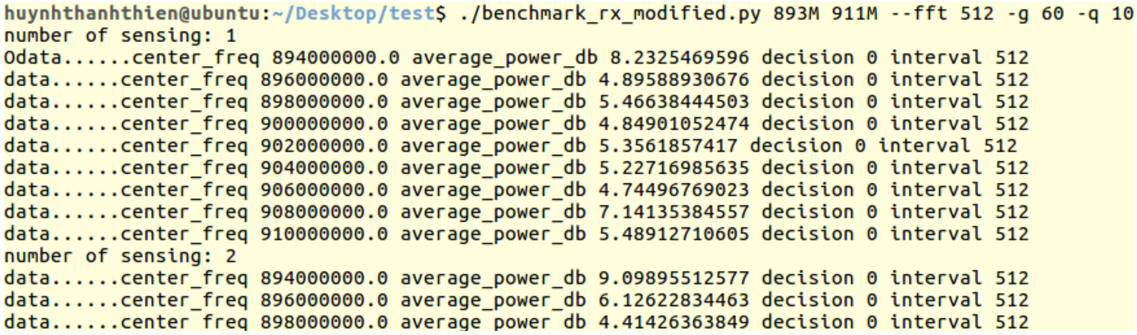


```
htt@ubuntu:~/Desktop/test$ ./benchmark_tx_modified.py 893M 911M --fft 512 -g 60 -q 4
number of sensing: 1
Odata ...........center_freq 894000000.0 average_power_db 6.13188746146 decision 1 interval 512
center_freq for transmit 896000000.0
Warning: failed to enable realtime scheduling
number of transmitting: 1
................................................................................................
....sent one file
begin retransmission
total data 65597
number of transmitting: 2
```

Figure 2.23: Spectrum sensing and video transmission at the transmitter with a threshold of 4 dB.

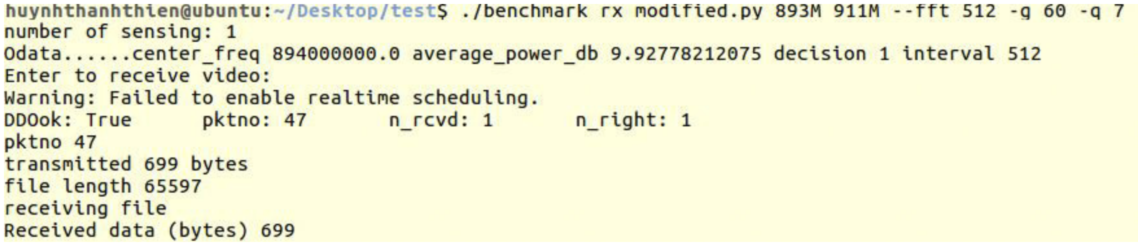
In Fig. 2.22, we can see that at a channel frequency of 894 MHz, the mean power (equal to 5.615 dB) is less than the threshold (8 dB), so the transmitter will transmit the video on the same channel. Conversely, in Fig. 2.23, if mean power (equal to 6.131 dB) is greater than the threshold (4 dB), the transmitter transmits the video on the next channel (a frequency of 896 MHz). At the receiver, the input of receiver Python script and the results of spectrum sensing on the receiver side with a threshold of 10 dB and 7 dB, are shown in Fig. 2.24 and Fig. 2.25, respectively.

In Fig. 2.24, we can see that the mean power on the channels is less than the threshold (10 dB), so the receiver will continue to sense the channel. Conversely, Fig. 2.25



```
huynhthanhthien@ubuntu:~/Desktop/test$ ./benchmark_rx_modified.py 893M 911M --fft 512 -g 60 -q 10
number of sensing: 1
Odata......center_freq 894000000.0 average_power_db 8.2325469596 decision 0 interval 512
data......center_freq 896000000.0 average_power_db 4.89588930676 decision 0 interval 512
data......center_freq 898000000.0 average_power_db 5.46638444503 decision 0 interval 512
data......center_freq 900000000.0 average_power_db 4.84901052474 decision 0 interval 512
data......center_freq 902000000.0 average_power_db 5.3561857417 decision 0 interval 512
data......center_freq 904000000.0 average_power_db 5.22716985635 decision 0 interval 512
data......center_freq 906000000.0 average_power_db 4.74496769023 decision 0 interval 512
data......center_freq 908000000.0 average_power_db 7.14135384557 decision 0 interval 512
data......center_freq 910000000.0 average_power_db 5.48912710605 decision 0 interval 512
number of sensing: 2
data......center_freq 894000000.0 average_power_db 9.09895512577 decision 0 interval 512
data......center_freq 896000000.0 average_power_db 6.12622834463 decision 0 interval 512
data......center freq 898000000.0 average power db 4.41426363849 decision 0 interval 512
```

Figure 2.24: Spectrum sensing at the receiver with a threshold of 10 dB.



```
huynhthanhthien@ubuntu:~/Desktop/test$ ./benchmark rx modified.py 893M 911M --fft 512 -g 60 -q 7
number of sensing: 1
Odata......center_freq 894000000.0 average_power_db 9.92778212075 decision 1 interval 512
Enter to receive video:
Warning: Failed to enable realtime scheduling.
DDOok: True      pktno: 47       n_rcvd: 1       n_right: 1
pktno 47
transmitted 699 bytes
file length 65597
receiving file
Received data (bytes) 699
```

Figure 2.25: Spectrum sensing at the receiver with a threshold of 7 dB.

shows that the mean power (9.927 dB) is greater than the threshold (7 dB), so the receiver downloads the video on the same channel (a frequency of 894 MHz).

2.5 Conclusion

In this chapter, we implemented a spectrum-sensing system with ED in a USRP NI 2900 and GNU Radio environment. Two main performance metrics for spectrum sensing were measured in real environments: the probability of false alarm P_{fa} and the probability of detection P_d. P_{fa} is used to determine the sensing threshold, whereas P_d is used to determine the desired performance. We observed that ED-based spectrum sensing can achieve higher performance when the number of sensing samples is larger. In addition, the results showed that evaluation of the probability of detection and the probability of false alarm can improve more when SNR is higher. Besides, in this chapter, ThingSpeak is also utilized to upload data to the cloud, which is measured in a local CR node, such that we can store, process, and share the sensed information more efficiently in a centralized way.

In addition, this chapter also presents the results of research into the implementation of USRP and GNU Radio for a spectrum-sensing system with video transmission. In this research, video in H.264 and MP4 formats can be transmitted on a specific channel or on a sensed channel. Performance of the video transmission is evaluated through packets error rate, which is based on the number of received packets compared to the number of transmitted packets. Performance on the testbed was evaluated based on exploring parameters, such as gain and distance between USRPs. We anticipate that gain and distance between two USRPs plays an important role. Results show that the transmission and reception of the video files is smooth and successful when the testbed is implemented at high gain and over a short distance. There could be distortion in the video as the distance increases due to a loss of packets or from packet errors.

Chapter 3

A Transfer Games Actor–Critic Learning Framework for Anti-Jamming in Multi-channel Cognitive Radio Networks

3.1 Introduction

Nowadays, the demand for communication and entertainment of users is increasing, leading to a significant increase in wireless applications and services. As a result, issues such as spectral scarcity and increasing demand for spectrum sources pose enormous challenges for network operators. To address existing issues, the CRN was developed [2, 22] and is considered one of the most promising technologies for improving spectrum efficiency. The basic idea of a CRN is to exploit spectrum holes by enabling SUs (also called unlicensed users) to sense, select, and access free channels which are not occupied by the PUs (also called licensed users). However, whenever a PU needs those channels, the SU has to vacate them.

For the implementation of efficient spectrum exploitation in CRNs, selection by users of the appropriate access channel has a great influence on the performance of the network. Many channel selection schemes have been investigated in the previous works [23–28]. Although the proposed solutions can utilized the spectrum effectively, they are all based on the assumption that SUs exploit spectrum holes, while coordinating together to achieve their common target. This assumption ignores the scenario in which different random attackers could attack communication channels between SUs that could threaten network security and interfere with CRN. Physical or media access control layers are vulnerable to attacks that are a security threat to communication in CRN. These threats are not only harmful to commercial networks but also threaten national defense and national security [29]. Hence, along with the challenges in spectrum management, secure spectrum utilizing also plays a crucial role for the development of CRN architecture. For that reason, a considerable amount of research on security techniques has been investigated for the CRNs [30, 31]. However, the influence of jammers on spectrum sharing has been still little considered. Some previous work proposed resource allocation and intelligent jamming to avoid security threats from jammers [32, 33]. In [34], the authors proposed an anti-jamming game in CRNs with multiple channels by modeling the interaction between a SU and attackers. Moreover, the anti-jamming game is redefined as the defense strategy with randomized power allocation. Most of the researchers only consider resource allocation and intelligent jamming to counter jammers attacks.

Recently, the Markov decision process (MDP) and the game theory approach in CRN have been investigated [3, 35, 36]. A stochastic game in [37] considered a competition and interaction among players, which is an extension of the MDP proposed in [38]. However, these proposed game approaches do not exploit the knowledge about the PU status on the channel, which can be collected via spectrum sensing on a pre-selected channel. In this work, we also solve the anti-jamming problem using the game theory approach. First, we propose a single-game scheme by formulating the problem of channel selection as a game framework, solving the problem for finding the best channel by using value iteration–based dynamic programming. In this chapter, anti-jamming means that there is the absence of PU on the channels under consideration and further the jammers are not accessing these channels. Subsequently, jammers do not jam PU but only jam on the channels of interest. Besides, jammers could be considered as malicious SUs that try to access channels to prevent other normal SUs from using them. Malicious SUs can forge the spectral characteristics of the PU to gain priority access to wireless channels, known as primary user emulation (PUE) attacks. To match with the scenario in this work, it is assumed that the CRN can easily detect PUE attacks based on several detection mechanisms such as channel parameters and spectrum decision, feature detection with filter and cyclostationary and many other detection mechanisms are mentioned [39]. Therefore, the jamming ability of attackers on PU is not considered in our work. Second, for the purpose of improving the performance of the single-game scheme through gathering the knowledge about the PU status on the channel, we can use the double-game scheme in our previous work [40]. In double-game scheme, the first game is solved to find pre-selected channel for the SU. Then, based on this channel, the SU performs spectrum sensing to collect PU status information on the channel. From the sensing results, the the belief and state of the system are updated and used for the second game to find the final best channel. Note that, our best channel selection problem against jammer's attacks is different from the previous work [40]. In this work, the problem of best channel selection against jammer's attacks is investigated in the scenario where communication channels are not used by the PUs and are not jammed by attackers. Meanwhile, the problem of best channel selection in [40] is based on maximize the secrecy rate of the SU.

Furthermore, the dynamic game solutions assume that the environment's dynamics (e.g., the jammer's strategies) is known in advance, which is rarely true due to the random attack nature of jammers. Since accurate information about the dynamics of the environment is sometimes not available, the problem of stochastic optimization is usually formulated as the MDP framework [41]. Afterward, the problem was formulated with MDP could be solved using RL approaches [42]. In RL, the agent makes the optimal policy through environmental interactions and requires no prior knowledge of the dynamic of the environment [43]. Because of the advantages of a RL approach, a series of studies have been carried out using the combination of anti-jamming and RL techniques [44–48]. Wang *et al.* [44] proposed an anti-jamming defense mechanism in CRN based on a stochastic game framework in which SUs can decide how many channels are used for a given purpose based on observations of the jammer's attack strategy, channel quality, and the spectrum availability. To learn the optimal policy, the spectrum-efficient throughput is maximized using the minimax-Q learning. Singh and Trivedi in [45] have proposed the anti-jamming approach using the State-action-reward-state-action (SARSA) and QV RL algorithm in which the SU can learn the jammer's strategy and the characteristics of the channel. The results show an improvement in the performance of QV and SARSA algorithm when compared with the minimax–Q learning algorithm. In these studies, the Q–learning algorithm is used for most of the anti-jamming mechanisms due to the advantage of not knowing the model of this algorithm. However, with high-dimensional or continuous inputs, anti-jamming problems can face challenges when using traditional Q–learning algorithms. Therefore, several algorithms have been proposed to overcome this weakness such as the deep Q–network (DQN) [46] and double DQN [47,48] which leverage a deep neural network to approximate the Q table. Specifically, Han *et al.* [46] proposed a two-dimensional anti-jamming mechanism for CRNs in which the SINR of the SU signals can be improved based on the exploitation of user mobility and spread spectrum. Besides, the anti-jamming scheme used a DQN-based approach to find the optimal policy of the network. The authors in [47] used the double DQN algorithm to counter the jammer in a multi-user manner with frequency hopping strategy attacks. Xu *et al.* [48] modeled the encounter between the jammer and the CRN based on a double DQN design to maximize the users' transmission rate. In this chapter, for a performance comparison with our proposed schemes, we can also solve the problem of channel selection to avoid jammer's attacks using an RL approach, called an actor–critic (AC) algorithm. Specifically, based on the state of the system, the long-term network performance is maximized to find the optimal channel policy that can be used against jammer's attacks.

In the case of RL, agents must get the information under a trial-and-error process to find an action in each state, because in the beginning they have no prior information on the environment [49]. Therefore, the procedure could take a considerable amount of time for learning in the AC algorithm to reach an optimal policy. To address this problem, we use transfer learning technique [50]. Regarding to transfer learning techniques, problems in target task can be effectively solved through the application of information obtained from source task [51]. Consequently, transfer learning has attracted a lot of interest from researchers [50–55]. Additionally, several studies of anti-jamming by combining RL and transfer learning have been investigated recently [56–58]. Chen *et al.* [56] proposed a RL-based power control scheme in which the WBAN coordinator and the in-body sensors can communicate with each other to defend against attacks. The Q-learning algorithm and the transfer learning method are used to obtain an optimal policy and accelerate the learning speed, respectively. Dai *et al.* [57] provided a safe version of deep RL for network security in which the risk level is estimated and the transfer learning technique is used to reduce initial random exploration. An anti-jamming scheme with the help of an unmanned aerial vehicle (UAV) in a cellular network is proposed in [58] where the deep RL algorithm is used to find the optimal relay policy. Furthermore, transfer learning is also used to help cellular networks battle jammers without knowing system models as well as observed communication states. In general, in the above-mentioned anti-jamming jobs, the RL algorithm is used quite commonly, however, these studies are either considered on basic wireless networks that are not CRNs or not considered combining with transfer learning technique. Therefore, the problem of anti-jamming by combining RL and transfer learning in CRNs is considered in this chapter. Furthermore, the transfer learning technique used to transfer knowledge from source task where the anti-jamming problem is solved based on game theory is also a highlight in this chapter. By using the learned knowledge about channel selection from historical period (the double-game period), the ongoing learning process can be accelerated in the target task during the classic AC period, and provide additional improvements to the channel selection problem. As a result, the problem of channel selection with the help of transfer learning technology is proposed based on the transfer of knowledge learned from double-game scheme into a classic AC algorithm, which is denoted as the Transfer Game-Actor-Critic (TGACT) scheme in this work.

In summary, the main contributions of this chapter are presented as follows:

- We investigate anti-jamming approaches for CRN with a multi-channel and multiple jammer, where an SU is transmitting data to a receiver SU while multiple jammers independently perform jamming on transmitter SU–receiver SU (SUtx–SUrx) transmissions. Each jammer attacks a random channel of interest. To optimize the security of a CRN, we propose an anti-jamming scheme by using game-theoretic concepts through definitions of states, actions, and players' rewards. The network scenario is modeled as a dynamic game, namely, a single-game scheme that finds the optimal channel for the SU in order to protect communication channels from jamming attacks. By using the optimal channel, the SU can receive maximum long-term reward which can reduce jammers' impact on channels. Then, we propose a double game–based anti-jamming scheme based on a repeat game algorithm in our previous work [40], which has demonstrated an improvement in performance compared to the single-game scheme. After that, the network performance with the proposed double-game scheme can be compared with a random-attack, single-game, and no-jammer schemes.
- Besides, the best channel selection problem with anti-jamming can be reformulated as an MDP framework. For a performance comparison with our proposed schemes, we consider the solution to the formulated MDP by using the classic AC algorithm, an RL approach where there is no need to know the jammers' access strategy in advance.
- Moreover, transfer learning technology is also applied (namely, the TGACT scheme),

which uses the transferred knowledge in the double-game period to accelerate the learning process and to provide performance improvements in channel selection, compared with a classic AC scheme and a transfer Actor-Critic (TACT) algorithm [54].

- To evaluate the performance of the proposed schemes, we use the average reward metric (also called the security level in this chapter) of the SU in different configurations. The simulation results show that the proposed schemes are quite resistant to jammer attacks, and achieve better performance compared to other conventional channel selection schemes. Specifically, the double-game scheme provides better performance in comparison with random-attack and single-game schemes. Furthermore, the performance of the proposed TGACT scheme is also better than the dynamic game, classic AC, and TACT schemes.

The remaining of this chapter is arranged as follows. Section 3.2 presents the system model and local spectrum sensing. Section 3.3 describes game formulation for channel selection with anti-jamming for single- and double-game schemes. The RL approach–based anti-jamming schemes are described in Section 3.4, which introduce the classic AC, the TACT [54], and the proposed TGACT schemes. In Section 3.5, we present the simulation results and discussions. Finally, Section 3.6 provides a conclusion.

3.2 The system model and local spectrum sensing

3.2.1 System model

Consider a CRN system where a transmitter SU tries to access the licensed channel of the PUs (K licensed channels) by using local spectrum sensing and send data to a receiver SU while jammers (E) independently perform jamming on a random channel of interest, as shown in Fig. 3.1. The channels are assumed to be an AWGN channel. Let $\mathcal{K} = \{1, \ldots, k, \ldots, K\}$ and $\mathcal{E} = \{1, \ldots, e, \ldots, E\}$ denote the set of channels and jammers, respectively. We assume that the SUtx always has data to transmit to the SUrx. For the convenience of tracking and formulation terminology, SU will be used instead of $SUtx$ in the remainder of this chapter. The operation of the system is in a time-slotted manner with slots of equal length and non-overlap, which are represented with the letter t. In this work, the operation of a PU on channel k is assumed to follow a two-state Markov discrete-time process. Let $S_{PU}(k) = \{A, P\}$ denote the PU state, in which the notations

A and P represent the absence and presence of the PU, respectively. The operation of Markov chain states of the PU is shown in Fig. 3.2, in which P_{PA} and P_{AA} represent the state transition probabilities between the two absence and presence states of the PU. Let S^k denote the PU state on channel k, and we define $\mathrm{P}^k_{AA} = \Pr\left(S^k_{(t+1)} = A|S^k_{(t)} = A\right)$ and $\mathrm{P}^k_{PA} = \Pr\left(S^k_{(t+1)} = P|S^k_{(t)} = A\right)$ as the transition probability of the PU from state A to itself and from state P to state A, respectively.

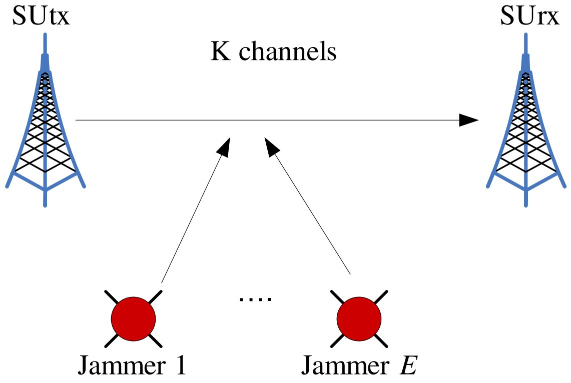


Figure 3.1: The system model.

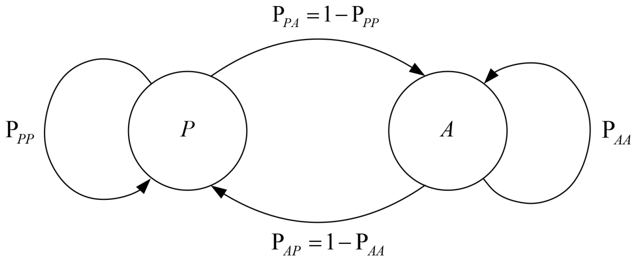

$\mathrm{P}_{PA} = 1 - \mathrm{P}_{PP}$
P_{PP}
P
A
P_{AA}
$\mathrm{P}_{AP} = 1 - \mathrm{P}_{AA}$

Figure 3.2: Markov chain for the PU states.

First, the SU perform local spectrum sensing on a particular channel by using energy detector. Then, the SU will perform the data transmission process on this channel when it is free. On the contrary, the SU is not allowed to occupy the channel for data transmission, and will wait until the next time slot to repeat the whole process.

In time slot t, the SU selects a channel for its communication, $x \in \mathcal{K}$, and a denotes the action of the SU with $a = \{x\,|x \in \mathcal{K}\}$. Action a will be a distribution over set $\mathcal{K}$, which is given as:

$$\mathrm{P}_{SU}(k) = \Pr\{a = k\} \tag{3.1a}$$
$$\text{s.t.:} \quad \sum_k \mathrm{P}_{SU}(k) = \sum_k \Pr\{a = k\} = 1, k \in \mathcal{K}, \tag{3.1b}$$

where $\mathrm{P}_{SU}(k)$ denotes the probability that the SU accesses channel k.

In the same way, the jammers select channels for jamming, $Y = \{y_1, y_2, \ldots, y_E\}, y_e \in \mathcal{K}$, and b_e denotes the action of jammer e where $b_e = \{y_e | y_e \in \mathcal{K}\}$. Then, actions by all jammers in the system are given as $b = \{b_1, b_2, \ldots, b_E\}$. Action b_e will be a distribution over set $\mathcal{K}$, which is given as:

$$\mathrm{P}_e(k) = \Pr\{b_e = k\} \tag{3.2a}$$
$$\text{s.t.:} \quad \sum_k \mathrm{P}_e(k) = \sum_k \Pr\{b_e = k\} = 1, k \in \mathcal{K}; e \in \mathcal{E}, \tag{3.2b}$$

where $\mathrm{P}_e(k)$ denotes the probability that jammer e attacks channel k.

Next, it is necessary to define the payoff function that characterizes the level of jamming. In this section, based on the exploitation of spectrum holes in CRNs, the payoff function is determined based on the characteristics of channel access behavior (related to user and jammers) and occupancy status of spectrum holes (related to PU), not the jamming intensity. Therefore, the payoff function is determined according to whether a particular channel is not under jammer's attacks and is not occupied by PU. Specifically, when the SU accesses channel k that is not under jammer's attacks, and this channel is not occupied by PU, the payoff function of the SU is given as:

$$R(a = k, b, S_{PU}(k)) = \begin{cases} 1, & \text{if } S_{PU}(k) = A \textit{ and } a \neq b_e, \forall e, \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

3.2.2 Local spectrum sensing

In the CRN considered, we assume the network includes a transmitter/receiver SU pair. The SU may use an energy detection method to perform local spectrum sensing. The binary hypothesis test of the SU is given as follows [59]:

$$\begin{cases} P: & x(t) = hu(t) + w(t), \\ A: & x(t) = w(t), \end{cases} \tag{3.4}$$

where $x(t)$ is the received signal by SU, h is the channel gain of the communication link between PU and SU, $u(t)$ is the signal transmitted by PU, and $w(t)$ is zero-mean AWGN. The obtained energy at the SU [60]:

$$yE = \sum_{j=1}^{I} \left| x\left(j\right) \right|^2, \tag{3.5}$$

where I is the number of sensing samples during each detection interval, and $x\left(j\right)$ is the received PU signal at the SU in the j^{th} sample. When I is adequately large (e.g., $I > 10$ in practice), we approximate yE as a Gaussian random variable under the binary hypothesis (P and A) with mean μ_P, μ_A and variance σ_P^2, σ_A^2 as [60]:

$$yE \sim \begin{cases} P: & \mathcal{N}\left(\mu_P = I\left(1+\phi\right), \sigma_P^2 = 2I\left(1+2\phi\right)\right), \\ A: & \mathcal{N}\left(\mu_A = I, \sigma_A^2 = 2I\right), \end{cases} \tag{3.6}$$

where ϕ is the sensed channel's SNR in decibels (dB). After that, two states of the PU can be made a decision as follows:

$$D\left(t\right) = \begin{cases} 1, & \text{when } yE\left(t\right) \geq \lambda, \\ 0, & \text{otherwise}, \end{cases} \tag{3.7}$$

where 0 and 1, respectively, are binary bits that denote two states of the PU, absence and presence; and λ denotes a predefined threshold of decision energy.

3.3 Game approach–based anti-jamming scheme

We model the problem of channel selection for the interaction between the SU and the jammers as a game framework by using game-theoretic concepts through definitions of states, actions, and players' rewards [61, 62]. The game formulation of channel selection problem to avoid jammer's attacks is represented as follows.

- **Players:** the number of players joining the game is $(1+E)$ players (i.e., an SU and E jammers).

- **State:** the system state is defined as:

$$\mathcal{S} = \left\{ \mathrm{P}_0\left(k\right), \mathrm{P}_e\left(k\right) | k \in \mathcal{K}; e \in \mathcal{E} \right\}, \tag{3.8}$$

where $\mathrm{P}_0(k)$ is the probability (also called the belief) that the state of channel k is A (i.e., not used by the PU). If we consider the operation of the SU, the state will be approximated as:

$$\mathcal{S}_{ch} = \{\mathrm{P}_{ch}(k) \,|k \in \mathcal{K}\}, \tag{3.9}$$

where $\mathrm{P}_{ch}(k)$ denotes the probability that the state of channel k ($\mathcal{S}_{ch}$) is free (i.e., not jammed and not being used by the PU), which is given as:

$$\mathrm{P}_{ch}(k) = \mathrm{P}_0(k)\left(1 - \prod_e \mathrm{P}_e(k)\right). \tag{3.10}$$

- **Action:** in each time slot t, the SU should select a channel for its communication, and a denotes the action of the SU with $a = \{x\,|x \in \mathcal{K}\}$.
- **Reward:** the reward for the SU, $\overline{R}(\mathrm{P}_{SU}(a), \mathcal{S}_{ch})$, is given by:

$$\overline{R}(\mathrm{P}_{SU}(a), \mathcal{S}_{ch}) = E[R(a, b,\, S_{PU}(a))] = \mathrm{P}_{SU}(a)\,\mathrm{P}_{ch}(a)\,R(a, -a,\, A), \tag{3.11}$$

where $E[R]$ denotes the expected value of the SU's payoff function R.

In this chapter, the goal of choosing the best channel for SU is to maximize long-term reward (also called the accumulated reward of the SU) of the system, $\overline{aR}\left(\mathrm{P}_{SU}(a), \mathcal{S}_{ch}^0\right)$, which is defined as follows:

$$\overline{aR}\left(\mathrm{P}_{SU}(a), \mathcal{S}_{ch}^0\right) = \sum_{t=m}^{\infty} \gamma^t \overline{R}\left(\mathrm{P}_{SU}(a), \mathcal{S}_{ch}^t | \mathcal{S}_{ch}^m = \mathcal{S}_{ch}^0\right), \tag{3.12}$$

where m is the current time slot, t is the t^{th} time slot, $\mathcal{S}_{ch}^m$ is the system state in time slot m, γ is a discount constant ($\gamma \in (0, 1)$).

Then, the problem of choosing the optimal channel for SU to protect communication channels from jammers attacks is identified as follows:

$$a_{opt} = \arg\max_a \left(\overline{aR}\left(\mathrm{P}_{SU}(a), \mathcal{S}_{ch}^0\right)\right). \tag{3.13}$$

3.3.1 Single game–based anti-jamming scheme

The problem in (3.13) can be solved by maximizing the accumulated reward of the SU. Through SU and jammers' channel access strategies, we can determine the state of the system. Therefore, the accumulated reward of the SU can be easily calculated based on its state action space. Besides, using a value iteration-based dynamic programming (DP) approach can obtain closed-form solutions for the value function [63–65]. Therefore, we consider our game model in which value iteration-based dynamic programming can be employed to come up with optimal strategies for the SU in order to find the optimal channel and protect communication channels from jamming attacks. The single game–based anti-jamming scheme is represented in Algorithm 3.1.

Algorithm 3.1 Single game–based anti-jamming scheme

```
**Input:** $K, E, \mathrm{P}_0(k), T, \mathrm{P}_{AA}, \mathrm{P}_{PA}, \gamma$.
**Output:** the optimal channel for the SU, $a_{opt}$.
Given the system state: $\mathcal{S} = \{\mathrm{P}_0(k), \mathrm{P}_e(k) \,|k \in \mathcal{K}; e \in \mathcal{E}\}$, as expressed in (3.8)
Determine the local decision for the state of the PU based on Section 3.2.2, local spectrum sensing.
**for** $t = 1$ *to* $T$
    **for** $a = 1$ *to* $K$
        Calculate:
        The payoff function: $R^t(a) \leftarrow (3.3)$
        The probability: $\mathrm{P}_{ch}^t(a) \leftarrow (3.10)$
        The reward for the SU: $\overline{R^t}\left(\mathrm{P}_{SU}^t(a), \mathcal{S}_{ch}^t\right) \leftarrow (3.11)$
    **end for**
    Calculate the accumulated reward: $\overline{aR}\left(\mathrm{P}_{SU}(a), \mathcal{S}_{ch}\right) \leftarrow (3.12)$
    Find the optimal channel, $a_{opt}^t$: $a_{opt}^t = \underset{a}{\arg\max}\left(\overline{aR}\left(\mathrm{P}_{SU}(a), \mathcal{S}_{ch}\right)\right)$
**end for**
```

3.3.2 Double game–based anti-jamming scheme

In this section, we propose a double game–based anti-jamming scheme for a CRN. The basic idea of the double-game scheme is to exploit the knowledge about the PU status on the channel [40], which can be determined by solving the best channel selection problem from the single-game scheme. Specifically, the double-game scheme utilizes single-game scheme two times. One is for pre-selected channel, and the other one is for the best channel after spectrum sensing. That is, we first choose a most preferable channel, which is called as the pre-selected channel by using the Algorithm 3.1. After performing spectrum sensing on the pre-selected channel, we choose the best channel using the Algorithm 3.1, again. The double-game scheme is composed of 4 phases of the first game, spectrum sensing, the second game and environment updating, which are shown in Fig 3.3. Specifically, the four phases of the proposed double-game scheme are presented as follows.

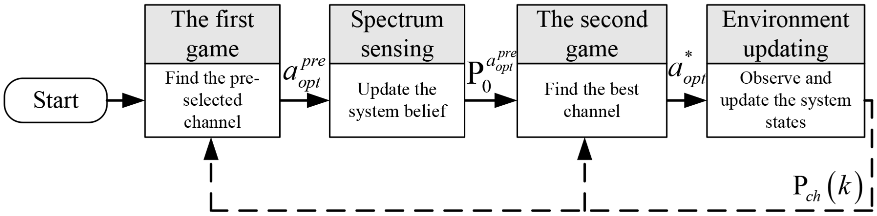


Figure 3.3: A block diagram of the proposed double-game scheme.

- **The first game phase (line 5 in Algorithm 3.2):** solving the problem in (3.13) based on initial state of the system to find the pre-selected channel, a_{opt}^{pre}.

- **Spectrum sensing phase (lines 6-8):** based on pre-selected channel, spectrum sensing is carried out by SU to exploit the knowledge about the PU status on channel. Depending on the sensing result for the state of the PU signal is A or P (i.e., the channel is free or busy, respectively), the system belief will be updated accordingly using Bayes' rule [66]. Specifically, if the channel is free, the belief, $\mathrm{P}_0^{a_{opt}^{pre}}$, is updated as follows:

$$\mathrm{P}_0^{a_{opt}^{pre}} = \frac{\mathrm{P}_0^{a_{opt}^{pre}} \left(1 - \mathrm{P}_f\left(a_{opt}^{pre}\right)\right)}{\mathrm{P}_0^{a_{opt}^{pre}} \left(1 - \mathrm{P}_f\left(a_{opt}^{pre}\right)\right) + \left(1 - \mathrm{P}_0^{a_{opt}^{pre}}\right)\left(1 - \mathrm{P}_d\left(a_{opt}^{pre}\right)\right)}, \tag{3.14}$$

where P_d and P_f are the probabilities of correct detection and false alarm, respectively. Otherwise, the belief is updated as follows:

$$\mathrm{P}_0^{a_{opt}^{pre}} = \frac{\mathrm{P}_0^{a_{opt}^{pre}}\, \mathrm{P}_f\left(a_{opt}^{pre}\right)}{\mathrm{P}_0^{a_{opt}^{pre}}\, \mathrm{P}_f\left(a_{opt}^{pre}\right) + \left(1 - \mathrm{P}_0^{a_{opt}^{pre}}\right) \mathrm{P}_d\left(a_{opt}^{pre}\right)}. \tag{3.15}$$

The state of the system is updated based on the estimated belief from either (3.14) or (3.15), which is denoted by $\mathcal{S}^u$. According to the updated state, $\mathcal{S}^u$, the system update the rewards and the accumulated rewards by (3.11) and (3.12), respectively.

- **The second game phase (line 9):** solving the problem in (3.13) based on the updated accumulated reward from spectrum sensing phase to find the final best channel, a_{opt}^{*}.

- **Environment updating phase (line 10):** the optimal reward can be obtained by using optimal channel of the SU, a^*_{opt}. Based on the observation of channel status, we need to update the system state for use in the next time slot. Specifically, if communication with channel a^*_{opt} is successful (i.e., the channel is not occupied by PU), the belief $\mathrm{P}_0^{a^*_{opt}}$ is updated as:

  $$\mathrm{P}_0^{a^*_{opt}} = \mathrm{P}_{AA}^{a^*_{opt}} . \tag{3.16}$$

  Otherwise, if communication fails (i.e., the channel is occupied by the PU), the belief is updated as:

  $$\mathrm{P}_0^{a^*_{opt}} = \mathrm{P}_{PA}^{a^*_{opt}} . \tag{3.17}$$

  The system state in (3.9) will be updated using the updated belief, $\mathrm{P}_0^{a^*_{opt}}$, for use in the next time slot.

In short, the double game–based anti-jamming scheme is represented in Algorithm 3.2.

Algorithm 3.2 Double game–based anti-jamming scheme

```
1: Input: $K, E, \mathrm{P}_0(k), T, \mathrm{P}_{AA}, \mathrm{P}_{PA}, \gamma$.
2: Output: the optimal channel for the SU, $a^*_{opt}$.
3: Given the system state: $\mathcal{S} = \{\mathrm{P}_0(k), \mathrm{P}_e(k) | k \in \mathcal{K}; e \in \mathcal{E}\}$, as expressed in (3.8)
4: Determine the local decision for the state of the PU based on Section 3.2.2, local spectrum sensing.
5: Find the optimal pre-selected action (the channel) of the game, $a^{pre}_{opt}$: $a^{pre}_{opt} = \arg\max_{a} \left(\overline{aR}\left(\mathrm{P}_{SU}(a), \mathcal{S}_{ch}\right)\right)$, which can be solved with Algorithm 3.1.
6: Implement spectrum sensing and update the belief about the system: $\mathrm{P}_0^{a^{pre}_{opt}} \leftarrow$ (3.14) or (3.15)
7: According to updated belief $\mathrm{P}_0^{a^{pre}_{opt}}$, update the state of the system: $\mathcal{S}^u \leftarrow$ (3.8)
8: Update the accumulated reward, $\overline{aR_u}$, based on the updated state, $\mathcal{S}^u$: $\overline{aR_u} \leftarrow$ (3.12)
9: Solve the problem in (3.13) with updated accumulated reward $\overline{aR_u}$ to find optimal channel $a^*_{opt}$ for the SU.
10: The optimal reward can be obtained by using optimal channel of the SU, $a^*_{opt}$. According to the observation of the communications link in the channel, update the system state for use in the next time slot by using (3.16) or (3.17).
```

3.4 Reinforcement learning approach–based anti-jamming schemes

We reformulated the best channel selection problem in a multi-channel CRN system as the framework of an MDP. Since the strategies of the jammers on communication channels are unknown, we employ the RL approach, which finds the optimal channel selection policy to reduce the jammer's influence and enhances the long-run network performance. In a model-free RL framework, RL agents can learn the optimal policy through trial-and-error learning during their interaction with the environment.

3.4.1 Markov decision process

A basic RL model is composed of two factors, environment and agent, in which these two elements interact over time. Furthermore, based on an environment states, the agent does a process of trial-and-error learning, and then the agent can make a suitable action and maximize the accumulated rewards. Regarding the MDP framework, we need to consider objects like state space ($\mathcal{S}_{ch}$), action space ($\mathcal{A}$), the state-transition probability function ($\mathcal{P}$), and the reward function (R). Therefore, the MDP framework of the channel selection problem for anti-jamming can be defined as a tuple $\langle \mathcal{S}_{ch}, \mathcal{A}, \mathcal{P}, R \rangle$.

- **States:** for the operation of the SU, the state is defined as $\mathcal{S}_{ch} = \{\mathrm{P}_{ch}(k) \,|k \in \mathcal{K}\}$, where $\mathrm{P}_{ch}(k)$ as defined in (3.10).

- **Actions:** at the time slot t, the agent observes state $\mathcal{S}_{ch}^{t}$ in state space $\mathcal{S}_{ch}$ of the environment, and then chooses action a^t in action space $\mathcal{A}$ following a probability of taking action, π. In this work, the SU (the network agent) chooses the best channel that it can access (i.e., the channel which is not occupied by the PU and not being jammed). Therefore, action a^t is set as $a^t = \{x\}_{x \in \mathcal{K}}$, which is defined as explained in Section 3.2.1.

- **Rewards:** then, the environment will return a reward to the agent, $R\left(\mathcal{S}_{ch}^{t}, a^t\right)$. The reward of the network can be defined as in (3.11), and transforms to the new state $\mathcal{S}_{ch}^{t+1}$. The next state, $\mathcal{S}_{ch}^{t+1}$, is updated following (3.9), which is based on the action (channel k).

- **The state-transition probability function:** once the SU selects an action, the system changes from the current state, $\mathcal{S}_{ch}^{t}$, to the new state, $\mathcal{S}_{ch}{}'$, based on the probability of state-transition as follows:

$$\mathcal{P}\left(\mathcal{S}_{ch}{}'|\mathcal{S}_{ch}^{t}, a^{t}\right) = \begin{cases} 1, & \text{if}\, \mathcal{S}_{ch}{}' = \mathcal{S}_{ch}^{t+1}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.18}$$

The purpose of the RL approach is to learn to select actions based on the states of the system through learning from experience to maximize the accumulated reward (also called the state-value function) of the system. The state-value function is expressed as follows [43]:

$$\begin{aligned} V\left(\mathcal{S}_{ch}\right) = \sum_{t=0}^{\infty} \gamma^{t} R\left(\mathcal{S}_{ch}^{t}, \pi\left(\mathcal{S}_{ch}^{t}\right) | \mathcal{S}_{ch}^{0} = \mathcal{S}_{ch}\right) \\ = R\left(\mathcal{S}_{ch}, \pi\left(\mathcal{S}_{ch}\right)\right) + \gamma \sum_{\mathcal{S}_{ch}{}' \in \mathcal{S}} \mathcal{P}\left(\mathcal{S}_{ch}{}'|\mathcal{S}_{ch}, \pi\left(\mathcal{S}_{ch}\right)\right) V\left(\mathcal{S}_{ch}{}'\right), \end{aligned} \tag{3.19}$$

where $\pi\left(\mathcal{S}_{ch}\right) : \mathcal{S}_{ch} \mapsto a$ denotes the stochastic policy which SU can take an action, a, based on the state of the environment, $\mathcal{S}_{ch}$, and $\mathcal{P}\left(\mathcal{S}_{ch}{}'|\mathcal{S}_{ch}, \pi\left(\mathcal{S}_{ch}\right)\right)$ denotes the state-transition probability from the current state $\mathcal{S}_{ch}$ to the next state $\mathcal{S}_{ch}{}'$. The Bellman equation is used to maximize the state-value function, and find the optimal policy, π^{*}, which is given as follows [43]:

$$\pi^{*}\left(\mathcal{S}_{ch}\right) = \arg\max_{a} \left(R\left(\mathcal{S}_{ch}, a\right) + \gamma \sum_{\mathcal{S}_{ch}{}' \in \mathcal{S}} \mathcal{P}\left(\mathcal{S}_{ch}{}'|\mathcal{S}_{ch}, a\right) V^{*}\left(\mathcal{S}_{ch}{}'\right) \right). \tag{3.20}$$

Through determining the optimal policy, we can find the optimal channel for SU which can avoid jamming from attackers in a multi-channel CRN.

3.4.2 The AC–based channel selection scheme

Traditionally, the MDP problem can be solved with a value iteration-based dynamic programming approach. However, this approach needs to know the dynamic environment in advance. In addition, the agent will face more challenges in the process of finding the optimal policy when using dynamic programming approach to solve the Bellman equation in a high-dimensional space of state and action. Therefore, for a performance comparison with our proposed schemes, we also consider using an RL approach, called the classic AC algorithm which requires no prior knowledge of the environment's dynamics. Regarding this approach, the agent can learn the optimal policy through trial-and-error learning during their interaction with the environment. Basically, an agent for the AC algorithm consists of two separate components [42]: the actor, which observes the environment state and selects an action by stochastic policy π; and the critic, which evaluate an actor's action based on the value function and reward, as shown in Fig. 3.4.

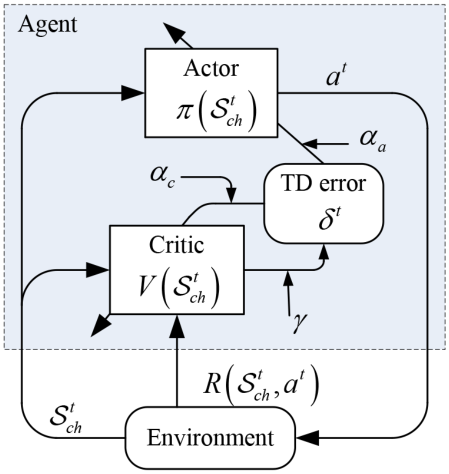


Figure 3.4: A block diagram of the classic actor–critic algorithm.

When an SU and jammers connect to the network, the initial state of the system is $\mathcal{S}$. In order to optimize performance and maximize the accumulated reward, the SU chooses suitable actions in which the selected channels are not used by the PU and are not being attacked by jammers. The learning process of the AC algorithm to find the optimal channel selection policy is presented as follows. In the time slot t, the SU selects an action, a^t, following a policy, $\pi^t\left(\mathcal{S}_{ch}^t\right)$. The probability of taking action a^t in state $\mathcal{S}_{ch}^t$ is given as follows [43]:

$$\pi^t\left(\mathcal{S}_{ch}^t, a^t\right) = \Pr\left(a^t \middle| \mathcal{S}_{ch}^t\right) = \frac{e^{h^t\left(\mathcal{S}_{ch}^t, a^t\right)}}{\sum\limits_{a'} e^{h^t\left(\mathcal{S}_{ch}^t, a'\right)}}, \tag{3.21}$$

where $h^t\left(\mathcal{S}_{ch}^t, a^t\right)$ is the tendency to select action a^t in state $\mathcal{S}_{ch}^t$. Once the SU selects action a^t, the current state, $\mathcal{S}_{ch}^t$, will transit to the next state, $\mathcal{S}_{ch}^{t+1}$, according to the state-transition probability, which is given in (3.18), and returns an immediate reward, $R\left(\mathcal{S}_{ch}^t, a^t\right)$. Afterward, based on the calculation of the temporal difference (TD) error value, the critic will evaluate the selected action from the actor. The TD error value is calculated from the value of $R\left(\mathcal{S}_{ch}^{t}, a^{t}\right) + \gamma V^{t}\left(\mathcal{S}_{ch}^{t+1}\right)$ at the critic and the state-value function in the previous state, $V^{t}\left(\mathcal{S}_{ch}^{t}\right)$, which is given as follows:

$$\delta^{t} = R\left(\mathcal{S}_{ch}^{t}, a^{t}\right) + \gamma V^{t}\left(\mathcal{S}_{ch}^{t+1}\right) - V^{t}\left(\mathcal{S}_{ch}^{t}\right). \tag{3.22}$$

Thereafter, based on the TD error, the critic will update its state-value function in the next time slot to improve the state-value function and policy. The state-value function is updated as follows:

$$V^{t+1}\left(\mathcal{S}_{ch}^{t}\right) = V^{t}\left(\mathcal{S}_{ch}^{t}\right) + \alpha_{c}\delta^{t}, \tag{3.23}$$

where α_c denotes the step-size parameter of the critic. Besides, the policy at the actor is also updated as follows:

$$h^{t+1}\left(\mathcal{S}_{ch}^{t}, a^{t}\right) = h^{t}\left(\mathcal{S}_{ch}^{t}, a^{t}\right) + \alpha_{a}\delta^{t}, \tag{3.24}$$

where α_a denotes the step-size parameter of the actor. Overall, the system performance can be improved by updating functions of the state-value and policy based on the TD error with appropriate step-size parameters by the actor and critic.

3.4.3 The TACT–based channel selection scheme

The previous section addresses the problem of finding the best anti-jamming channel using the classic AC algorithm. In this section, we present a methodology where the controller utilizes information on the strategies learned during the historical period to find the best anti-jamming channel. First, *state, action, reward* and *value function* definitions are also defined as described in Section 3.4.1. For a performance comparison with our proposed schemes, a TACT algorithm in [54] can be applied to our channel selection problem. The block diagram of the TACT scheme is shown in Fig. 3.5. For a TACT-based approach, the information on the policy, $h\left(\mathcal{S}_{ch}, a\right)$, from a source task (left side in Fig. 3.5) is transferred to a target task (right side in Fig. 3.5). However, there might be some differences although the target task and the source task have similarities. For example, the source task has a higher reward than the target task even though these two tasks use the same state. Therefore, action a can be taken by the controller in the target task in an aggressive direction for channel selection. As a result, the transferred policy can have a negative effect on the action selection process. Hence, by reducing the effects of the transferred policy, we can mitigate these negative effects. In general, the basic idea of the TACT algorithm is to avoid

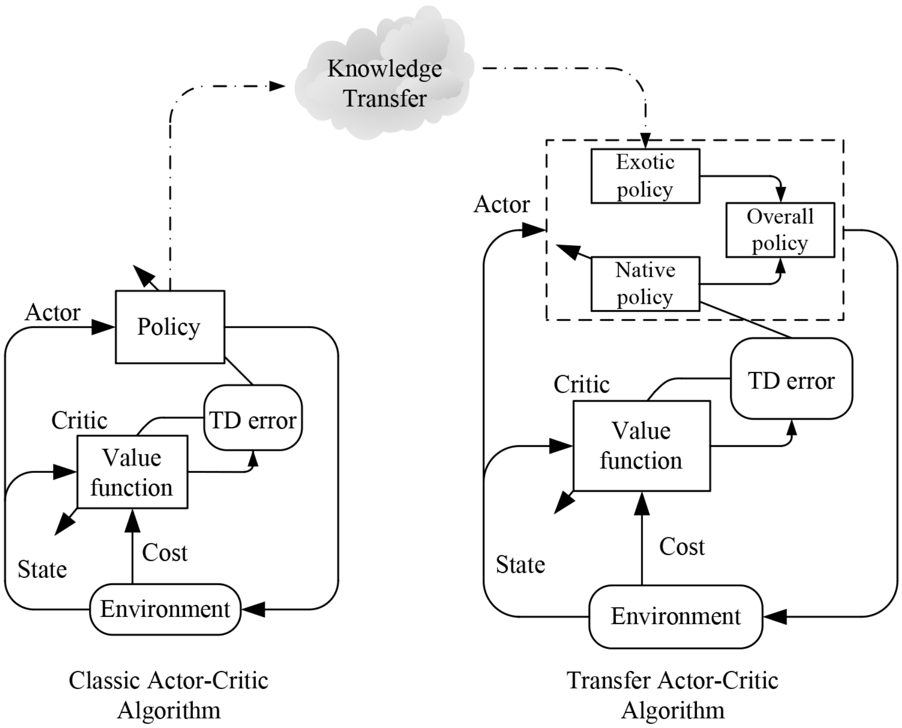


Figure 3.5: A block diagram of the TACT scheme [54].

the negative effect of the transferred policy on the action selection process [54]. From this idea, we can update the overall policy with transfer rate where the transfer rate should be decreased over time to reduce the impact of transferred policy on the overall policy. As can be seen in Fig. 3.5, the overall policy, h_o, is a combination of an exotic policy (also called transferred policy), h_e, and a native policy, h_n. The overall policy is updated as follows [54, 55]:

$$h_o^{t+1}\left(\mathcal{S}_{ch}^t, a^t\right) = (1-\zeta)\, h_n^{t+1}\left(\mathcal{S}_{ch}^t, a^t\right) + \zeta h_e\left(\mathcal{S}_{ch}^t, a^t\right), \tag{3.25}$$

where $\zeta \in (0, 1)$ denotes the transfer rate which represents the exotic policy contribution to the overall policy. During the initial training process, the overall policy update strategy with the dominance of the exotic policy over the native policy, so the performance of the system can be improved. However, the goal is still learning at the target task, so we need to reduce the impact of transferred policy on the overall policy. Therefore, the transfer rate should be decreased over time with decay factor d_ζ, and thus, $\zeta \mapsto 0$ as the number of iterations reaches infinity. Besides that, the native policy updates itself according to the classic AC algorithm as defined in (3.24).

3.4.4 The proposed TGACT–based channel selection scheme

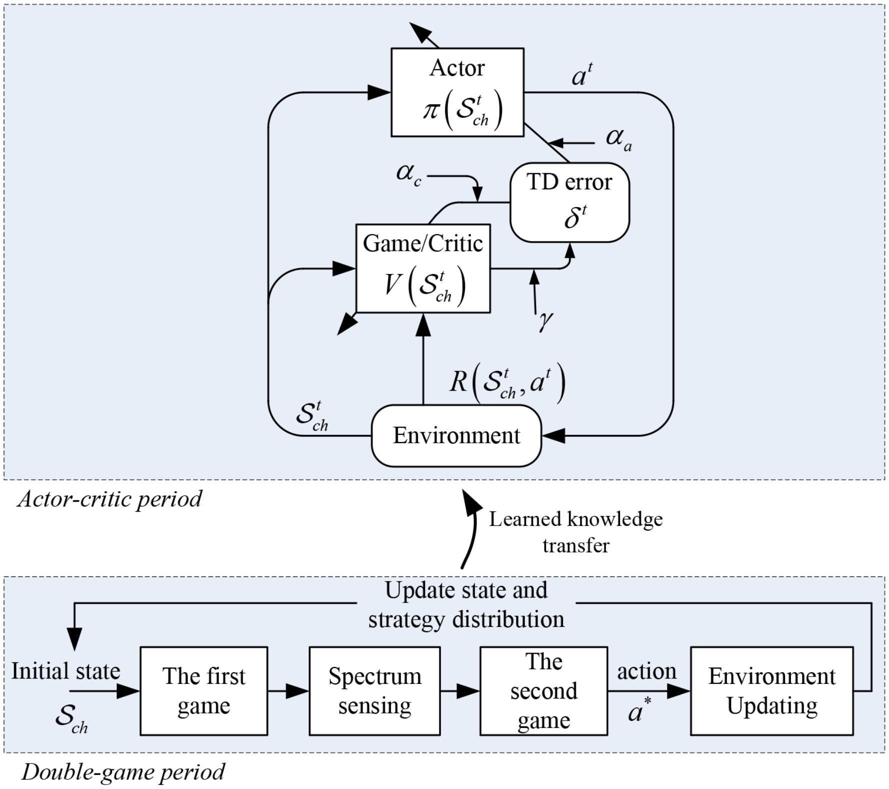


Figure 3.6: A block diagram of the transfer Game–AC scheme.

Transfer learning method in the TGACT–based channel selection scheme consists of two phases: i) transferring information from using the optimal channel, which can be obtained from the double-game period; and ii) training the target task based on the updated state and strategy distribution. More specifically, in the first phase of transfer learning, the Algorithm 3.2 is exploited to get the knowledge about the PU status on the channel to find the optimal channel. The communication link status on this channel can then be determined to be either occupied or not occupied by the PU. As a result, the PU state, P_0, should be updated according to (3.16) and (3.17) and will be transferred to the second phase of transfer learning. In the second phase, the learning process is implemented by using the classic AC algorithm as described in Section 3.4.1 and Section 3.4.2 with the updated state, P_0, from the first phase. The TGACT block diagram is shown in Fig. 3.6, and the proposed TGACT-based anti-jamming scheme is presented in the Algorithm 3.3 in which the first phase of transfer learning is from line 3 to line 6 and the learning process of the classic AC algorithm is from line 7 to line 16.

Algorithm 3.3 The proposed TGACT–based anti-jamming scheme

```
1: **Input:** $K, E, \mathrm{P}_0(k), T, \mathrm{P}_{AA}, \mathrm{P}_{PA}, \gamma, \alpha_c, \alpha_a, \zeta, d_\zeta$.
2: **Output:** the optimal channel selection policy, $\pi^*_{opt}$.
3: Determine the local decision for the state of the PU based on Section 3.2.2, local spectrum sensing.
4: Find the optimal channel for the SU, $a^*_{opt}$, which can be determined with Algorithm 3.2.
5: Based on the optimal access channel, update the state of the system as seen in (3.16) or (3.17).
6: Determine the system state in (3.8) based on the updated state.
7: Initialize the lookup table for policy $\pi(\mathcal{S}_{ch}, a)$, tendency $h(\mathcal{S}_{ch}, a)$, and state-value function $V(\mathcal{S}_{ch})$.
8: **for** $t = 1$ *to* $T$
9:      Select action $a^t$ based on temporal policy $\pi^t(\mathcal{S}^t_{ch}, a^t)$
10:     Calculate immediate reward: $R(\mathcal{S}^t_{ch}, a^t) \leftarrow (3.11)$
11:     Update the state of the system from $\mathcal{S}^t_{ch}$ to $\mathcal{S}^{t+1}_{ch}$, and calculate TD error: $\delta^t \leftarrow (3.22)$
12:     Update the state-value function: $V(\mathcal{S}^t_{ch}) \leftarrow (3.23)$
13:     Update the tendency to select an action: $h^{t+1}(\mathcal{S}^t_{ch}, a^t) \leftarrow (3.24)$
14:     Update the policy: $\pi(\mathcal{S}^t_{ch}, a^t) \leftarrow (3.21)$
15: **end for**
16: Return the optimal policy: $\pi^*(\mathcal{S}_{ch}) = \underset{a \in \mathcal{A}}{\arg\max}\{\pi(\mathcal{S}_{ch}, a)\}$
```

3.5 Simulation results and discussion

In this section, we show simulation results to demonstrate the efficiency of the proposed schemes for anti-jamming in multi-channel CRNs. We also compare the performance of our proposed schemes, which include single-game, double-game, and TGACT schemes, against the performance of other baseline schemes, such as the classic AC scheme, the TACT scheme [54], a random-attack scheme, and a no-jammers scheme. We sometimes use terms like learning and non-learning. The learning schemes include classic AC, TACT, TGACT, and double-game schemes. The double-game scheme is seen as an improved method of the single-game scheme; it exploits the action in the current time slot to update the system belief when using in the next time slot. Therefore, we consider the double-game scheme as one of the learning schemes. The non-learning schemes include the single-game, the no-jammers, and the random-attack schemes. With a single-game scheme, in the current time slot, the player selects an action and tries to maximize the accumulated reward of the system. When using the no-jammers scheme, there are no jammer attacks on the channels. With the random-attack scheme, jammers randomly attack the channels, and there are no anti-jamming efforts applied to the system.

3.5.1 Simulation settings

In this section, we compare the performance of the proposed schemes under various configurations. First, we show the convergence property for each of the proposed schemes in terms of the average rewards metric, ave_R, which are calculated as follows:

$$ave\_R = \frac{1}{T}\sum_{t=1}^{T} Re\left(t\right), \tag{3.26}$$

where T is the number of time slots, $Re\left(t\right)$ is the reward for the SU in the t^{th} time slot which is calculated by (3.11) without effect of the channel selection probability of SU. Then, we validate the network performance in terms of average rewards under three conditions: varying the number of channels, varying the number of jammers, and varying SNR value of the sensed channel. In the first scenario, simulations are performed when the number of jammers was fixed at $E = 5$ while the number of channels changed from three to 11. In the second scenario, we consider the performance of the proposed schemes when the number of channels is $K = 5$ while the number of jammers changes from one to five. The SNR of the sensed channel is $\phi = -6$ dB in both first and second scenarios. In the last one, the SNR of the sensed channel changes from -18 dB to -2 dB, while the number of jammers and the number of channels are each fixed at 5. In all cases, we assume that the initial values of the state transition probabilities are $\mathrm{P}_{AA} = 0.8$ and $\mathrm{P}_{PA} = 0.2$. The value of discount factor, $\gamma = 0.99$. To provide the best performance from the proposed schemes, the simulations are performed several times to achieve the most suitable step-size parameters (α_a and α_c). Then, we set $\alpha_a = 0.1$ and $\alpha_c = 0.1$. As seen in previous work [54, 55], higher transfer rates resulted in a faster convergence rate and better performance. Therefore, the transfer rate is set to $\zeta = 0.9$ with a decay rate of $d_\zeta = 0.99$. Simulations are performed with $T = 2,000$ timeslots. All of the schemes are implemented using Matlab.

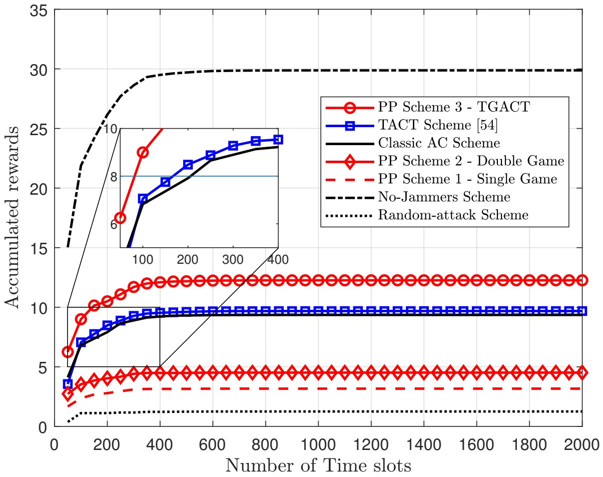


Figure 3.7: Accumulated rewards with five channels ($K = 5$) and five jammers ($E = 5$) when the SNR of the sensed channel is -6 dB ($\phi = -6$ dB).

3.5.2 Convergence property

In this section, we check the convergence property in terms of the accumulated rewards from our proposed schemes when the number of time slots, T, increases gradually from 1 to 2,000. The number of channels and jammers are fixed at $K = E = 5$, while the SNR of the sensed channel is set to $\phi = -6$ dB. Fig. 3.7 shows the improvement of the accumulated reward as the number of time slots increases. We observe that the accumulated rewards from the schemes increase rapidly over the first 400 time slots, and reach optimal value with more time slots. The convergence speed in the game schemes is faster than the random-attack scheme with no anti-jamming. The convergence speed of the double-game scheme is faster than the single-game scheme owing to information exploitation about the PU status on the channel that is collected via sensing on the optimal channel obtained from the single-game scheme. However, with the single-game scheme, the selected action is not affect the future reward, the player only try to choose the action in the current time slot which maximize the accumulated reward of the system. Therefore, the convergence speed in dynamic programming schemes like the random and single-game schemes is much lower than in RL schemes like the AC scheme and the transfer learning schemes (TACT and our proposed TGACT). The convergence speed of transfer learning–based RL schemes is faster than the AC scheme owing to the advantage of transfer learning, which transfers information from the source task to the target task. Fig. 3.7 also shows that schemes using transfer learning (TGACT and TACT) can accelerate learning process of conventional RL algorithm, AC scheme. Specifically, to reach the value of the accumulated rewards of 8, the TGACT and TACT schemes need 70 and 160 time slots, respectively. Meanwhile, to get this value of the accumulated rewards, the classic AC scheme needs about 210 time slots. From this result, we can see transfer learning can accelerate RL process.

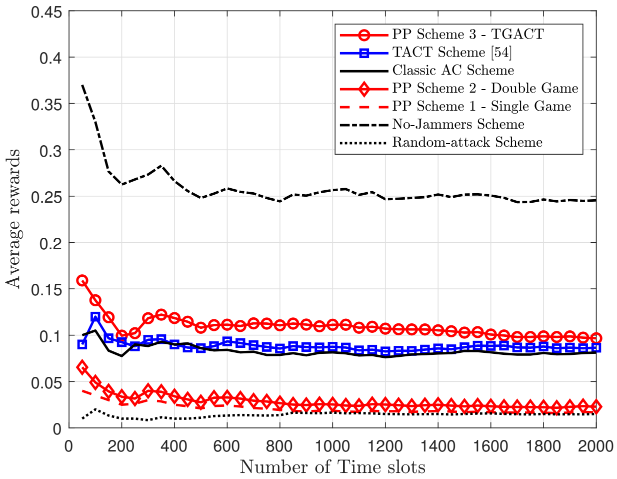


Figure 3.8: Average rewards with five channels ($K = 5$) and five jammers ($E = 5$) when the SNR of the sensed channel is -6 dB ($\phi = -6$ dB).

Afterward, we verify the convergence property of the proposed schemes in terms of the average reward. As seen in Fig. 3.8, the convergence rate of the schemes significantly decreases over the first 200 time slots, then, the average reward continues to decrease but at a slower rate. Finally, the schemes reach to an optimal reward to use for channel selection after about 1,000 time slots. The reward for the SU that used the classic AC scheme is lower than for SUs using the transfer learning schemes. This is because the agent of the transfer learning schemes can learn faster by exploiting transferred knowledge from source task. In addition, the agent in classic AC scheme needs to be trained from scratch, and therefore, it needs more trials-and-errors to learn. The convergence rate of the TGACT scheme outperforms in most learning schemes. The random-attack scheme provides the lowest convergence speed, and thus, got the smallest rewards. For most of the schemes, the performance is the best in a favorable environment with no jammer attacks on the system. In the convergence process, if the agent uses too many time slots for training, a local optimal policy might be obtained. However, the training process might take a very long time. Therefore, the total number of time slots for training should be neither too large nor too small.

3.5.3 The performance of the system according to the number of channels, the number of jammers, and the SNR of the sensed channel

In Fig. 3.9, we observe the performance of the proposed schemes under the influence of the number of channels. In this case, the number of channels is set at $K \in \{3, 5, 7, 9, 11\}$ while the number of jammers and the SNR of the sensed channel are fixed at $E = 5$ and $\phi = -6$, respectively. As seen in Fig. 3.9, the average reward increases as the number of channels increases. In fact, the more channels are used, the weaker the ability to attack a particular channel, and thus obtain higher system rewards. The average reward from the single-game scheme dominated the random-attack scheme. To explain this, with the single-game scheme, the SU maximizes the accumulated reward based on the action selection at the current time slot, whereas there are no anti-jamming solutions used in the system with the random-attack scheme. The double-game scheme is better than the single-game scheme owing to exploitation of PU status information, which can be collected via sensing based on the optimal channel from the single-game scheme. Moreover, the average reward of the proposed TGACT scheme outperforms the classic AC and TACT schemes. In particular, when the number of channels is five, the average reward of the TGACT scheme provides improvements of 10.51 % and 15.94 % over TACT and classic AC schemes, respectively. This is because, in the proposed TGACT scheme, agent can exploit information transferred from double-game period, and thus, learn effectively the optimal policy. Therefore, the TGACT scheme provides the best performance in comparison with the remaining schemes, except for the no-jammers scheme. With the no-jammers scheme, system performance is the best compared to most other schemes. However, in this scheme, jammers are not allowed to attack the channels. Moreover, although the number of channels changes, the local decision for the state of the PU is specified only once for the initial parameter. Hence, the system reward from this scheme remains unchanged. The average reward is lowest in case of the random-attack scheme. This is because the SU does not use channel selection schemes with anti-jamming and channels can be randomly attacked by jammers.

Fig. 3.9 also shows the Kullback-Leibler (KL) divergence [67] in which KL diver-

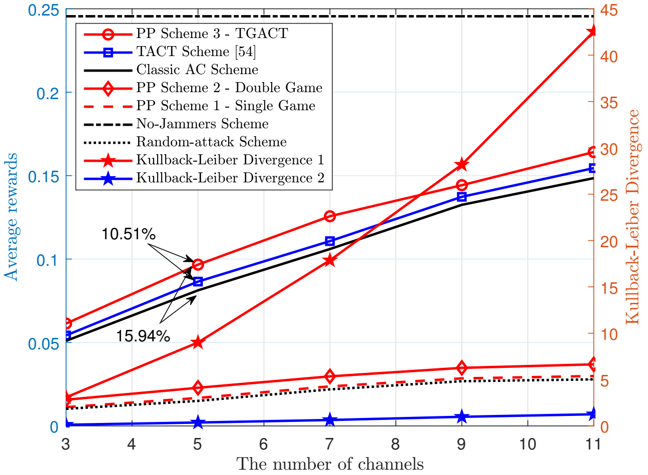


Figure 3.9: The left Y-axis shows average rewards according to the number of channels when the number of jammers is $E = 5$ and the SNR value of the sensed channel is $\phi = -6$ dB. The right Y-axis represents the Kullback-Leibler (KL) divergence in which KL divergence 1 represents the KL divergence of TGACT scheme over Double-game scheme and KL divergence 2 represents the KL divergence of TACT scheme over classic AC scheme.

gence 1 represents the KL divergence of TGACT scheme over Double-game scheme and KL divergence 2 represents the KL divergence of TACT scheme over classic AC scheme. Comparing KL divergence 1 and KL divergence 2, the implementation of a transfer learning from a double-game scheme to classic AC scheme (i.e., TGACT scheme) provides a significant improvement in performance over performing a transfer learning from an AC scheme to AC scheme (i.e., TACT). Furthermore, the properties of KL divergence show that the smaller the DL divergence value, the more similar the two distributions are. Therefore, with a small improvement in average rewards, the KL divergence 2 achieves a relatively low gain when the TACT scheme and the classic AC scheme are compared. Likewise, KL divergence 1 with a significantly large value can be explained. As a result, the higher KL divergence between the target task and the source task, the more efficient it is in performing transfer learning. The simulation results also show that the average rewards improve with increasing the number of channels, the value of KL divergence also increases.

In the same way, we evaluate the efficiency of proposed schemes under varying numbers of jammers, E, and compare the results with the classic AC, the TACT, the random-attack, and the no-jammers schemes, as shown in Fig. 3.10. While the number of jammers ranges from one to five, the number of channels and the SNR of the sensed channel are fixed at $K = 5$ and $\phi = -6$ dB, respectively. When the number of jammers is high, the channel becomes more vulnerable to attack due to the large number of jammers. Therefore, with an increase in the number of jammers, the average reward in the system decreases significantly. In addition, the performance of proposed schemes is dominant than the conventional channel selection schemes because the channel can be selected effectively by maximizing the system reward in the current time slot, as in the game schemes, or by learning the variations in the environment and transferring information from double-game period, as done in the TGACT scheme. Again, the system performance is also remained unchanged in no-jammers scheme because jammers are not allowed to attack the channels in this scheme.

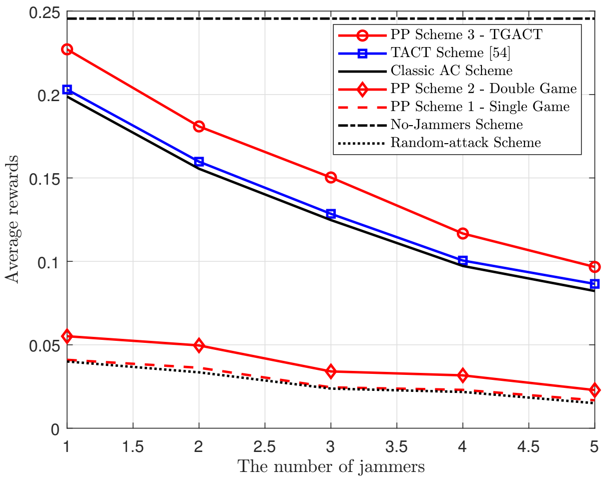


Figure 3.10: Average rewards according to the number of jammers when the number of channels is $K = 5$ and the SNR of the sensed channel is $\phi = -6$ dB.

We further inspect the impact of the SNR of the sensed channel on the security level of the channel selection schemes, which is shown in Fig. 3.11. To verify this, we evaluate the results based on the following SNR values (in decibels), $\phi \in \{-18, -14, -10, -6, -2\}$, while keeping the number of channels and jammers at $K = 5$ and $E = 5$, respectively. As observed in Fig. 3.11, the achieved average reward increases with an increase in the SNR of the sensed channel, which enables SU to effectively spectrum sensing and local decision-making. Obviously, a better SNR provides better detection accuracy. The result is that the larger SNR of the sensed channel may provide a better overall performance. Again, the TGACT scheme provides the highest average reward, whereas the random-attack scheme shows the lowest average reward. This is because the TGACT scheme is able to choose the effective channel in each time slot by the combination of estimating the future reward and the exploitation of the transferred information from double-game period. Meanwhile, the random-attack scheme does not use a channel selection scheme against jammer's attacks to enhance the security level. Consequently, we verify that the TGACT scheme can provide effective communication channels in terms of security level.

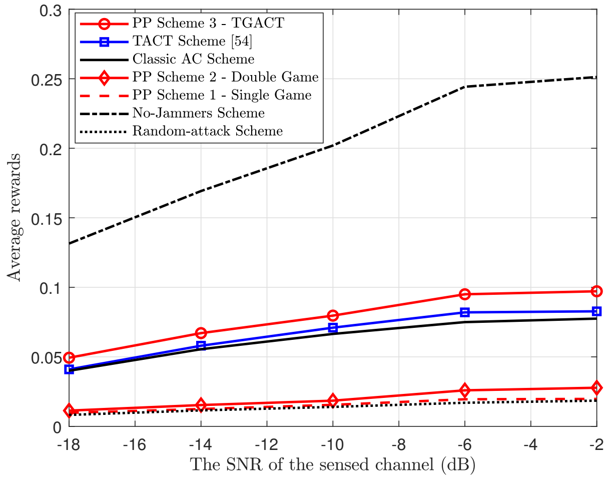


Figure 3.11: Average rewards according to the SNR of the sensed channel when the number of channels and jammers are $K = 5$ and $E = 5$, respectively.

3.6 Conclusions

In this work, we proposed anti-jamming approaches for a CRN in which the SU works multi-channel communications, and various numbers of jammers randomly attack. We first designed a single game–based anti-jamming scheme that solves the problem of maximizing the accumulated reward for the SU in order to find the optimal channel. Then, a double game–based anti-jamming scheme is considered, in which the pre-selected channel is determined by using a single-game scheme. Afterward, through the pre-selected channel, the SU performs spectrum sensing to collect the PU status information, then, the second game will be solved using the updated accumulated reward. In addition, we adopted the transfer learning technique into the double-game scheme to accelerate the learning speed and improve network performance by exploiting the information learned in the double-game period. The simulation results show the efficiency of the proposed solutions in improving the long-term performance of the network. Through the proposed schemes, the optimal channel will be provided for SU to avoid jamming from attackers and significantly improve the security level of the CRN. The channel selection problem with anti-jamming can be extended with multiple SUs in the future work. Consequently, the model and learning parameters would need to be modified. Even the state and action spaces will be larger, and thus, the problem becomes more complicated. For this reason, combining both transfer learning technology and a deep RL approach could be considered, in which deep neural network can be used as an approximation function for mapping the system input (e.g., the system state) and the output in the RL task (e.g., the optimal policy).

Chapter 4

Deep Learning-Based Approach to Fast Power Allocation in SISO SWIPT Systems with a Power-Splitting Scheme

4.1 Introduction

Recently, with the development of the Internet of Things (IoT), the requirements for high spectral efficiency (SE) and energy efficiency (EE) are higher and higher [68]. In communication systems, different popularly used technologies are either designed for sending information on communication channels of mobile devices (including cellular networks, wireless LANs, and Bluetooth) or are designed for power transfer (such as Qi and WiTricity) [69]. Moreover, RF has been regarded as a potential resource for EH in wireless systems. Since the information and energy are contained simultaneously in RF signals, SWIPT is presented for the first time in [70], and is regarded as a potential technology, where both information and energy of the common transmit signal are transmitted to the receivers [5, 71]. Therefore, SWIPT has engaged a lot of attention in the research community. SWIPT schemes for SISO system have been investigated [4, 70, 72], where the time-switching (TS) scheme and power-splitting (PS) scheme were considered. The study of SWIPT in interference channels is of more practical interest than in non-interference channels. Therefore, many studies on

SWIPT addressing the interference channel (IFC) have been done [73–81].

Two practical architectures at receiver in SWIPT systems are the TS receiver and the PS receiver [71, 81]. The TS scheme at the receiver switches between information decoding (ID) and EH over time, while the PS scheme splits the received signal into two parts, one is implemented for the ID and the other is done for EH. The SWIPT system with a PS scheme has been investigated for solving the minimum transmit power problem [75, 76]. In this chapter, we investigate a SWIPT system with multiple single-antenna transmitters and multiple single-antenna receivers. The studied scenario is based on the PS structure. By optimizing transmit power and PS ratios, the power optimization problem is studied under considering of the requirements for SINR and EH. The system with the PS scheme is similar to the SWIPT system that has been studied in [75], but is studied in a SISO SWIPT scenario with IFC. Moreover, our purpose is to open up new research directions with a combination of the two areas of DL and optimization problems by taking advantage of DL to solve the optimization problems.

In wireless communication systems (WCSs), the management and allocation of wireless resources like transmit power must be properly executed in order to obtain high network performance. For example, UEs manage transmit power inefficiently, which can cause a large amount of interference among UEs, and further result in a decrease in network performance. Therefore, the proper allocation of wireless resources is becoming more important with the increase in the popularity of UE density in WCSs. In this work, we also study power problem in WCSs but in a SWIPT system, and we are not too focused on optimization algorithms.

In previous researches, transmit power was obtained by solving iterative optimization problems, which had a specified set of network condition parameters such as channel gains and SINR requirements as their input. Those studies ran an entailed number of iterations before achieving convergence, and produced the optimized resource allocation strategy as their output. The iterative optimization algorithms have been efficiently solved the associated resource allocation problems and achieved relatively high performance. However, the iterative nature of these algorithms increases the computation time, which may lead to long latency and high computation costs in real-time operations due to the significant changing of some network parameters, so the entire iterative procedure of the algorithms has to be re-executed. Therefore, real-time operation of transmit power control still faces much difficulties in practical use. The problem becomes more serious when the number of more iteration is essential as the number of users increases [82]. For instance, the weighted minimum mean square error (WMMSE) algorithm needs complicated operations like the inversion of matrix [83, 84], singular value decomposition [85, 86] in each iteration, and semidefinite relaxation [87, 88].

Recently, using DL technology has been investigated in many domains, e.g., image processing, and it has proven superior performance compared to conventional schemes [89–92]. In particular, DL has started to be dealt with wireless communication systems, e.g., communication signals classification [90], channel estimation and signal detection [92], indoor localization [93], sparse optimization [94], and the optimization of constellation mapping [95]. Besides, through DNNs, DL technology can be used to efficiently solve sophisticated nonlinear problems through back-propagation algorithms [82] with low computation time [92] without the need to derive complex mathematical models. The authors in [82] approximated the transmit power based on WMMSE that was studied in [83] using a dense neural network, which can solve the main shortcoming of the WMMSE-based algorithm, for example, a long computation time caused by a huge number of iterations. Moreover, the validity of using a DNN in practical systems of wireless communication fields was confirmed in [96] using a testbed.

Unlike conventional resource management schemes in which sophisticated optimization problems have to be dealt with an iterative manner, we propose a DL-based approach for wireless resources allocation (in particular, for transmit power management) that can be obtained with low computation time. The basic idea is to use a set of wireless resources that is solved by an optimization algorithm, and to attempt to learn its input and output by using DL technologies based on DNNs or RNNs [97]. If a network efficiently approximates to a wireless resource allocation optimization algorithm, then the input of the optimization algorithm is passed into the trained network to get the output with higher computational efficiency in comparison with the optimization algorithm. This is because its testing stage does not involve iterative optimization; it only needs some layers and simple operations like matrix-vector multiplication/addition and simple nonlinear transformations. Therefore, if the training stage can efficiently approximate the optimization algorithm, the computation time for resource allocation in real time can be reduced significantly. Overall, our approach can be regarded as using a DL technology which based on DNNs and RNNs models for approximating an iterative optimization algorithm according to given network parameters.

The main contributions of this chapter are summarized as follows.

- We investigate a SWIPT system with multiple single-antenna at transmitters and receivers with a PS structure. The information and energy are transmitted simultaneously from transmitters to receivers. The transmit power optimization problem subjects to required SINR and EH is investigated by optimizing transmit power and the PS ratio at the receiver.

- We propose a DL-based approach to the transmit power optimization problem in SISO interference channels with SWIPT systems. The proposed approach opens up new research directions with a combination of the two fields of DL and wireless resources management (specifically, the combination of DL architectures and the transmit power optimization problem over interference channels). In the proposed approach, we use DL architectures including a type of DNN: the FFNN and three types of RNN: the LRN, the NARX, and LSTM. To the best of our knowledge, this is the first attempt to apply DL technology based on the RNN for transmit power control in the SWIPT system.

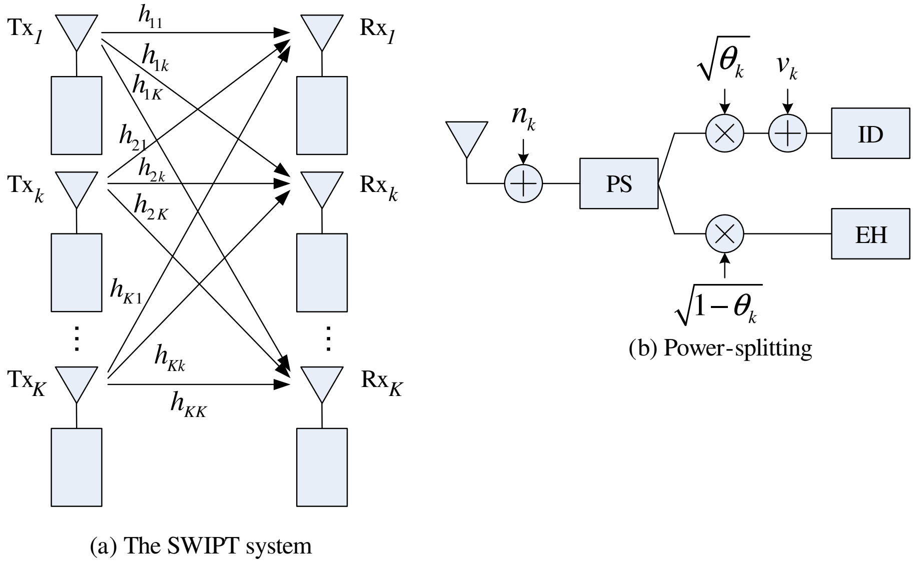
- Through our proposed approach, the transmit power of transmitters, and the PS ratio of the receivers can be obtained with lower complexity and less computation time, compared to conventional iterative approaches. Simulation results show that the DL-based approach is a great potential tool for approximating iterative optimization algorithms.

The rest of this chapter is arranged as follows. Section 4.2 shows the system model, and the problem formulation and solution for the SWIPT system. The DL-based proposed approaches are described in Section 4.3. The numerical results and discussions are provided in Section 4.4. Finally, Section 4.5 provides a conclusion.

4.2 System Model, Problem Formulation and the PS Solution

4.2.1 System Model

In this chapter, we investigate the transmit power optimization problem of SWIPT system. Consider a SWIPT system comprising of K transmitter-receiver (Tx-Rx) pairs with single-antenna on all transmitters and receivers where the k-th Tx-Rx pair is denoted by Tx_k–Rx_k ($k \in \mathcal{K} = \{1, \ldots, K\}$), as shown in Fig. 4.1. It is assumed that the transmitter transmits an omnidirectional signal, and a receiver, Rx_k, can collect the signal from transmitter Tx_k and all the signals from other transmitters Tx_j $(j \in \mathcal{K}, j \neq k)$, which interference with the target signal. Because the received signal could not be used for ID and EH at the same time, every receiver is connected to a time switcher when utilizing the TS scheme, or to a power splitter when utilizing the PS scheme. In this work, we are not too focused on the optimization problem, but only on the solutions that can help improve the complexity and computation time in comparison with the conventional optimization algorithm. Therefore, to reduce the system complexity, we just investigate the PS scheme.



(a) The SWIPT system

(b) Power-splitting

Figure 4.1: The SWIPT system model with a PS scheme.

Let h_{kk} denotes the direct channel of the $Tx_k - Rx_k$ pair. At the transmitter Tx_k, we assume that the transmitted symbol is a Gaussian random variable with zero mean and variance p_k (which is regarded as the transmit power at Tx_k). Moreover, assume that the transmitted symbols of different transmitters are independent. The transmitted signal is $x_k = \sqrt{p_k} s_k$ where s_k is the symbol which carries information at the transmitter Tx_k with $E\left\{|s_k|^2\right\} = 1$.

We assume that receivers under the PS scheme can concurrently implement ID and EH from received signals as shown in Fig. 4.1b. From PS structure, receiver Rx_k can divide the received signal into an ID stream with power ratio $\theta_k \in (0, 1)$ and an EH stream with power ratio $(1-\theta_k)$. The received signal at receiver Rx_k is:

$$r_k = \sum_{j=1}^{K} \sqrt{p_j} h_{jk} s_j + n_k, \quad \forall k = 1, \ldots, K, \tag{4.1}$$

where $n_k \sim \mathcal{CN}\left(0, \sigma_k^2\right)$ is AWGN from the antenna at the receiver Rx_k. The SINR at Rx_k is given as:

$$\mathrm{SINR}_k = \frac{\theta_k p_k |h_{kk}|^2}{\theta_k \sum\limits_{j \in K \backslash \{k\}} p_j |h_{jk}|^2 + \theta_k \sigma_k^2 + \delta_k^2}, \quad \forall k, \tag{4.2}$$

where $v_k \sim \mathcal{CN}\left(0, \delta_k^2\right)$ is the circuit noise at receiver on ID stream. The harvested energy at receiver is given by:

$$\mathrm{EH}_k = \mu_k \left(1-\theta_k\right) \left(\sum_{j=1}^{K} p_j |h_{jk}|^2 + \sigma_k^2 \right), \quad \forall k, \tag{4.3}$$

where $\mu_k \in (0,1]$ is the energy harvesting efficiency from the EH of Rx_k.

4.2.2 Problem Formulation and PS Solution

In this work, our target is to minimize the sum of the transmit power of the transmitters by optimizing the PS ratios and the transmit power subjects to the required SINR and harvested energy. Then, we formulate the power optimization problem as follows:

$$\underset{\{p_k\},\{\theta_k\}}{\text{minimize}} \quad \sum_{k=1}^{K} p_k \tag{4.4a}$$

$$\text{s.t.:} \quad \frac{\theta_k p_k |h_{kk}|^2}{\theta_k \sum\limits_{j \in K \backslash \{k\}} p_j |h_{jk}|^2 + \theta_k \sigma_k^2 + \delta_k^2} \geq \gamma_k, \quad \forall k \tag{4.4b}$$

$$\mu_k \left(1-\theta_k\right) \left(\sum_{j=1}^{K} p_j |h_{jk}|^2 + \sigma_k^2 \right) \geq e_k, \quad \forall k \tag{4.4c}$$

$$0 < \theta_k < 1, \quad \forall k \tag{4.4d}$$

$$p_k > 0, \quad \forall k. \tag{4.4e}$$

In constraint (4.4b), in order to guarantee the information services of receivers, the received SINR is greater than the minimum required SINR, γ_k. Constraint (4.4c) guarantees providing the harvested energy at the receiver. The problem (4.4) is changed to (4.5):

$$\underset{\{p_k\},\{\theta_k\}}{\text{minimize}} \quad \sum_{k=1}^{K} p_k \tag{4.5a}$$

$$\text{s.t.:} \quad -\left(\frac{1}{\gamma_k}+1\right) p_k |h_{kk}|^2 + \sum_{j=1}^{K} p_j |h_{jk}|^2 + \sigma_k^2 + \frac{\delta_k^2}{\theta_k} \leq 0, \quad \forall k \tag{4.5b}$$

$$\frac{e_k}{\mu_k\left(1-\theta_k\right)} - \left(\sum_{j=1}^{K} p_j |h_{jk}|^2 + \sigma_k^2\right) \leq 0, \quad \forall k \tag{4.5c}$$

$$0 < \theta_k < 1, \quad \forall k \tag{4.5d}$$

$$p_k > 0, \quad \forall k. \tag{4.5e}$$

4.3 The Deep Learning-Based Approaches

Although optimization algorithms can achieve relatively high performance in solving the associated resource management problems, these algorithms entail a huge number of iterations, which increase the computation time. So, the implementation of these algorithms in real time is still a challenging issue. In this section, we suggest a DL-based approach to wireless resources allocation, where wireless resources can be approximated by DL architectures such as the DNN and RNNs. The proposed DL-based approach is shown in Fig. 4.2. The basic idea is to use a set of wireless resources that is solved by an optimization algorithm, and to attempt to learn its input and output through the training stage. If a network efficiently approximates to a wireless resource allocation optimization algorithm, then the input of the optimization algorithm is passed into the trained network to get the output with higher computational efficiency.

4.3.1 Network Structure

This section describes the DL-based approaches used for minimum transmit power and PS ratio prediction, and presents a kind of DNN, the FFNN, and three kinds of RNN: the LRN, NARX, and LSTM.

An FFNN is an artificial neural network where the connections between the units do not form a directed cycle or loops, which is different from an RNN. The information passes through the network follow one direction, from the input layer through the hidden layers and to the output layer [98].

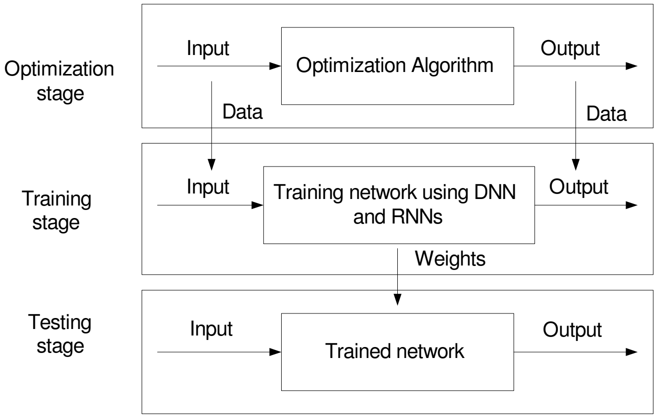


Figure 4.2: The proposed DL-based approach.

The RNN is a technology of learning in the DL, which has been attracting attention in recent years. RNN performs a memory mechanism with a recurring structure. Through this structure, neurons can use the information of the past in order to affect the output at the current moment, which is suitable for the prediction of time series data. In this work, three kinds of RNN are used: the LRN, NARX, and LSTM.

An earlier simplified version of the LRN was introduced by Elman [99]. The LRN is a type of RNN that has a single delay and a feedback loop at each hidden layer, but not at the last layer. The basic LRN architecture is constituted by three layers: input layer, hidden layer, and output layer, as presented in [100].

NARX is a dynamic RNN whose feedback loop is connected from the last layer to the input layer. The NARX model is based on the linear autoregressive exogenous model, which is used in time-series modeling. The formulation and the basic architecture for NARX are given in [101, 102].

Unfortunately, the training process of traditional RNNs is influenced by the exploding gradient, which can cause learning to diverge, or the vanishing gradient, where learning either becomes very slow or stops working altogether [103, 104], which prevents complete learning of the time series. One solution is to utilize LSTM networks. Therefore, we have also investigated LSTM RNNs, which introduces a new structure called a memory cell, as presented in [105, 106].

4.3.2 Optimization Stage

In the optimization stage, we treat the required SINR and the required harvested energy as fixed constants, and generate a large number of channel realizations $\{|h_{ij}|\}$ following certain distributions (specified in Section 4.4). Each tuple $(\gamma_k, e_k, \{|h_{ij}|\})\,; i, j \in \mathcal{K}$ is then fed to optimization algorithms like problem (4.5), and obtains the corresponding optimized power and PS ratio vectors $\{p_k, \theta_k\}$. The functions $\frac{\delta_k^2}{\theta_k}$ and $\frac{e_k}{\mu_k(1-\theta_k)}$ are convex with variable θ_k where $\theta_k \in (0, 1)$. So, Equations (4.5b) and (4.5c) are also convex function. Therefore, the minimum power problem is a convex optimization problem and Matlab's CVX [107] can efficiently solve this problem.

4.3.3 Training Stage

After solving the minimum transmit power problem in the optimization stage, we obtained optimal power and PS ratio vectors $\{p_k, \theta_k\}$ corresponding to channel vectors $\{|h_{ij}|\}$ with $i, j \in \mathcal{K}$ and $\mathcal{K} = 1, ..., K$. Then, we treat $(\{|h_{ij}|\} \,/\, \{p_k, \theta_k\})$ as an input/output pair, and try to learn the input/output relation through the training process of the DL-based approaches. Assuming that we use N samples for training data, then, we have input/output pairs $\left(\left\{\left|h_{ij}^n\right|\right\} / \{p_k^n, \theta_k^n\}\right)$ where $n = 1, \ldots, N$.

With the simple structure of an FFNN, we can use the input/output pairs as above, i.e., $\left(\left\{\left|h_{ij}^n\right|\right\} / \{p_k^n, \theta_k^n\}\right)$, for the training stage of the DNN. The training data is trained by using back propagation, and the training stage is implemented based on optimizing the mean squared error (MSE) by using the scaled conjugate gradient algorithm (which is both memory and computationally efficient [108]).

In the LRN and NARX, when the data are concurrent (matrix format), we need to change it into sequential data (a cell array format) before setting the network parameters because the input appears in time order. The LRN and NARX have a delay in the feedback loop. The feedback loop affects the order of the input, and the output. Therefore, we need to add more data corresponding to the number of delays used, i.e., $\left(\left\{\left|h_{ij}^T\right|\right\} / \left\{p_k^T, \theta_k^T\right\}\right)$, where T is the number of delays in each feedback loop. Then, we define input and output pairs for the training stages of the LRN and NARX as follows: $\left(\left\{\left|h_{ij}^T\right|; \left|h_{ij}^n\right|\right\} / \left\{p_k^T, \theta_k^T; p_k^n, \theta_k^n\right\}\right)$.

The LRN and NARX are trained by using the scaled conjugate gradient algorithm. After training stage, the MSE can be calculated. With a NARX neural network, firstly, the network is trained in an open loop, like the FFNN, with the scaled conjugate gradient algorithm. After that, the trained network with the open loop is changed to a closed loop in which target values are replaced by feedback signals from the output. Finally, the network is retrained in close loop form.

Although LSTM works with sequence and time series input data for data prediction, like the LRN and NARX, this network does not have the delays in the input and each feedback loop. Therefore, we also use $\left(\left\{\left|h_{ij}^{n}\right|\right\} / \left\{p_k^n, \theta_k^n\right\}\right)$ as input/output data for training. The LSTM network is trained by using the Adam optimization algorithm to update network weights, instead of using the classical stochastic gradient descent method.

Since the DNN and RNN architectures use backpropagation, we need activation function that calculates its own derivative. In this work, the following activation functions for hidden layers and output layer are used and given by (4.6) and (4.7), respectively.

$$tansig\left(n\right) = \frac{2}{1+e^{-2n}} - 1, \tag{4.6}$$

$$purelin\left(n\right) = n. \tag{4.7}$$

4.3.4 Testing Stage

In this stage, we also generate the channels $\left\{\left|h_{ij}^{m}\right|\right\}$, where $m = 1, \ldots, M$ with M as the number of samples for testing. The channels follow the same distribution as the training stage. Data preparation for the testing stage is the same process as the training stage for the FFNN, LRN, and NARX, and for the LSTM network. Then, testing data with channel vectors is run through the trained network.

4.4 Numerical Results and Discussions

In this section, we provide numerical results to showcase the effectiveness of the proposed approach. We first describe the simulation setup and neural network parameter selection, then demonstrate the capability of the proposed DL-based approaches that can produce responses similar to those produced by the optimization algorithm in changing the required SINR and the required harvested energy values. Neural network parameter selection is based on evaluation of the system performance by computing the MSE and efficiency of the computation time. In this work, in order to calculate the parameters of the networks such as weights and biases of each layer, we use stochastic gradient descent. The stochastic gradient descent picks a randomly chosen subset of data by which neural network is trained with gradient approximation. To compute gradients, we use the back-propagation algorithm (BP). The optimal weight and bias are obtained by minimizing MSE between the actual and desired output values [109]. Besides, in order to quantitatively compare the model fitting error, the MSE is a typical metric for evaluating the accuracy of the DL-based model [110–112]. The MSE is a measure of prediction accuracy which is calculated between observed values, y_i, taken from the optimization algorithm and the predicted values, $\widehat{y_i}$, in the testing stage taken from DL methods, $MSE = \frac{1}{n}\sum\limits_{i=1}^{n}(y_i - \widehat{y_i})^2$. The smaller the MSE value, the higher the predicted performance. The computation time in DL-based approaches includes two categories: training and testing times. Meanwhile, the computation time of the optimization scheme is attained according to the number of samples of the testing stage.

4.4.1 The Simulation Setup and Neural Network Parameter Selection

In our simulation, the proposed approaches are implemented in Matlab R2018b on a computer with 16 GB of memory and a 3.40 GHz CPU. For the parameters in the optimization problem, we set the Gaussian antenna noise at $\sigma_k^2 = -90$ dBW, and the circuit noise at $\delta_k^2 = -60$ dBW. Assuming that the required harvested energy e_k and the required SINR γ_k for all receiver are equal. The EH efficiency at the receiver is assumed to be the highest one such that $\mu_k = 1$. A standard normal distribution is used to generate channel coefficients; for example, we use a Gaussian distribution with zero mean and unit variance. The SWIPT system operates with two transceiver pairs ($K = 2$). To gain further insight into the impact of the DL-based approach, and to take real-time computational efficiency into account, we use a simple FFNN and three types of RNN: the LRN, NARX, and LSTM. We use a relatively large training data set with 10,000 samples ($N = 10,000$), and used 1000 samples ($M = 1000$) as a testing data set. The proposed DL-based approaches include an input layer, two hidden layers, and an output layer. In the neural network parameter selection process, the number of considered neurons for hidden layers is 20, 40, and 60. For the LRN and NARX, the number of delays on input is set to zero, and the number of delays in each feedback loop is set to 1 ($T = 1$). The initial values for the required SINR and harvested energy set at 6 dB and -20 dBm, respectively. The network parameter selection process can strike a good value between the MSE and the computation time. For the LSTM network, we set some options for training. In particular, the gradient threshold is 0.01, and the maximum number of epochs is 1000. To reduce the amount of padding in the mini-batches, we chose a mini-batch size of 20. Since the mini-batches are small with short sequences, training is better suited to the CPU, so we specified the environment for execution as CPU. The learning rate is set to 0.001.

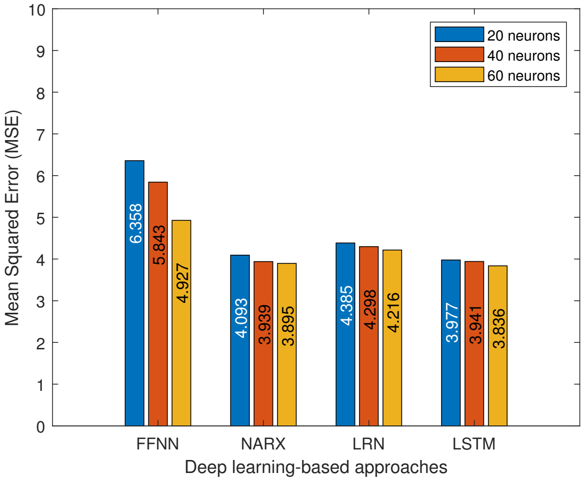
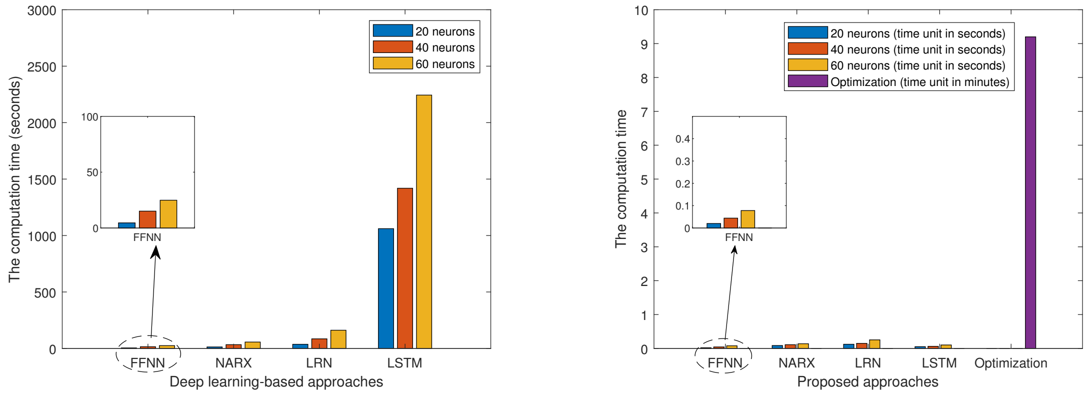


Figure 4.3: The MSE of DL-based approaches in the testing stage when the size of hidden layer is 20, 40, and 60 neurons, respectively.

Fig. 4.3 shows the network performance in term of the MSE in the testing stage. As we increase the hidden layers size from 20 neurons to 60 neurons, the complexity of the network increases, and the MSE might decrease. However, the value of MSE varies slightly. The FFNN provides a bad performance in term of MSE. This is because FFNN often encounters issues like overfitting or underfitting, which makes predictions for new data be inaccurate. Therefore, the FFNN does not fit the data efficiently. In the case of the LRN approach, the MSE tends to increase, compared to NARX and LSTM networks. This is due to the feedback loop structure in each hidden layer of the LRN network (i.e., the more hidden layers are used, the more loops exist in the network). As a result, the MSE of LRN network might increase. Since LSTM network can solve the gradient vanishing and exploding problems in traditional RNNs, the LSTM network provides the lowest MSE as shown in Fig. 4.3.

Fig. 4.4 shows the computation time of both training and testing stages where 10,000 samples are used for training in the FFNN, NARX, LRN and LSTM, and 1000 samples are used for testing when the hidden layer size changes from 20 neurons to 60 neurons. In all

DL-based approaches, the training and testing time are the lowest when 20 neurons are used in each hidden layer. In Fig. 4.4a, the computation time of the LSTM approach increases dramatically compared to other DL-based approaches. This is because LSTM has more backpropagation neural network modules, and the more memory information is transmitted over time, the more the cell state increases. Fig. 4.4a also shows that the computation time increases when the hidden layer sizes increase. In Fig. 4.4b, the testing time in DL-based approaches is very small, but the computation time of the optimization-based scheme is significantly larger. Therefore, we present the computation time of the optimization scheme in minutes in order to facilitate observation. Overall, the computation time of the DL-based approaches is much lower when compared to the optimization algorithm-based approach. This is easily explained because the testing stage does not involve iterative optimization; it only needs some layers and simple operations like matrix-vector multiplication/addition and simple nonlinear transformations. Therefore, if the training stage can estimate the optimization algorithm well enough, the computation time for resource allocation in real-time can be reduced significantly.



(a) Training stage

(b) Testing stage

Figure 4.4: The computation time of training stage and testing stage when the size of hidden layer is 20, 40, and 60 neurons, respectively: (**a**) Computation time among FFNN, NARX, LRN and LSTM in the training stage. (**b**) Computation time among FFNN, NARX, LRN, and LSTM and optimization scheme in the testing stage.

In addition, we also tested network performance by increasing the number of hidden layers in the selection process of neural network parameter in terms of MSE and computation

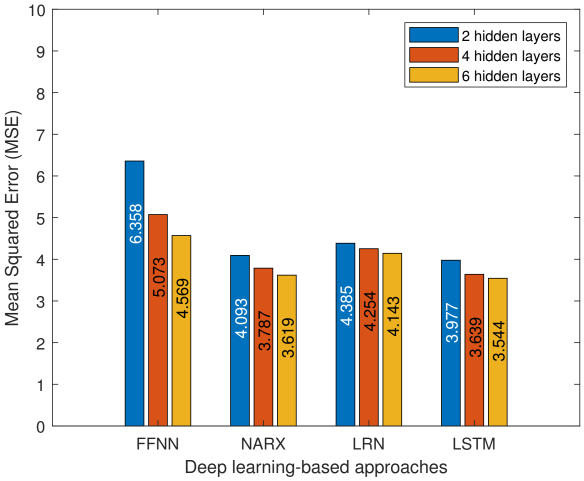
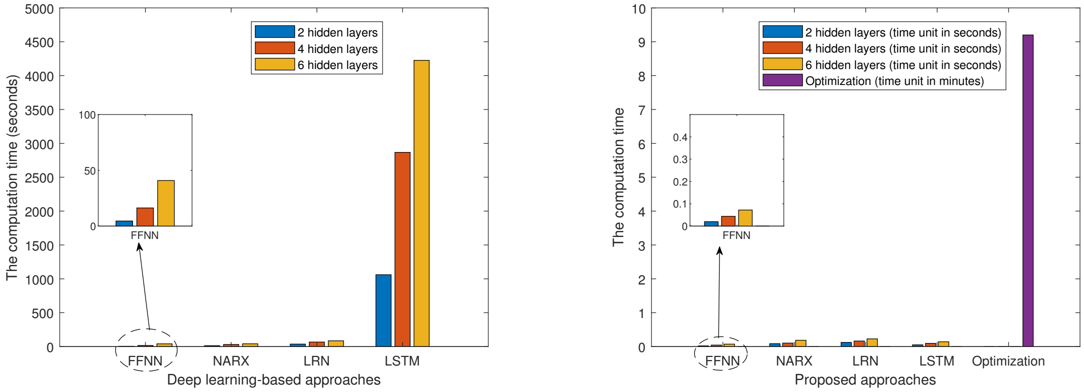


Figure 4.5: The MSE of DL-based approaches in the testing stage when the number of hidden layers is 2, 4, and 6, respectively.



(a) Training stage

(b) Testing stage

Figure 4.6: The computation time of training stage and testing stage when the number of hidden layers is 2, 4, and 6 layers respectively: (**a**) Computation time among FFNN, NARX, LRN and LSTM in the training stage. (**b**) Computation time among FFNN, NARX, LRN, and LSTM and optimization scheme in the testing stage.

time when networks used 20 neurons in the hidden layers, as shown in Fig. 4.5 and Fig. 4.6. Fig. 4.5 shows that the MSE value changed very slightly among different schemes when the number of hidden layers is increased from 2 to 6. Fig. 4.6 shows the computation time of both training and testing stages where 10,000 samples are used for training in the FFNN, NARX, LRN and LSTM, and 1000 samples are used for testing. In the testing stage, we measured the computation time of the optimization scheme to get 1000 samples, for the purpose of computation time comparison between DL approach and optimization approach. Fig. 4.6a shows that the computation time of all DL networks increases very slightly in the training stage as the number of hidden layers increases. Among DL networks, LSTM has the largest computation time. Figure 4.6b also shows that the computation time of all DL networks increases very slightly in the testing stage as the number of hidden layers increases. However, it is noteworthy that all DL networks have very small amount of computation time, compared to that of optimization scheme.

From the simulation results in Fig. 4.3, Fig. 4.4, Fig. 4.5, and Fig. 4.6, we have chosen 20 neurons and two hidden layers for the next simulation results with which we can get the best balance in the tradeoff between MSE and computation time.

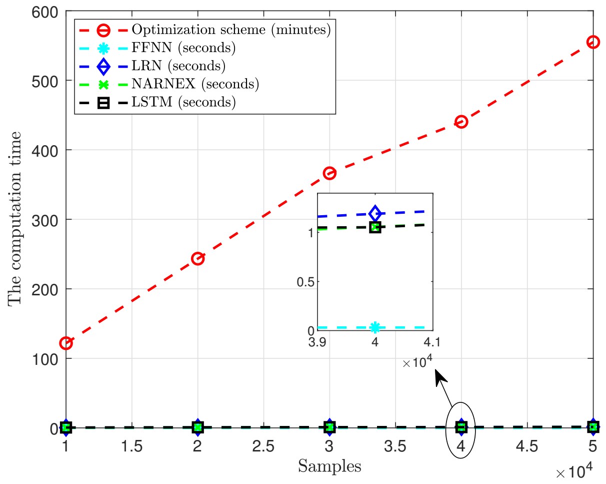


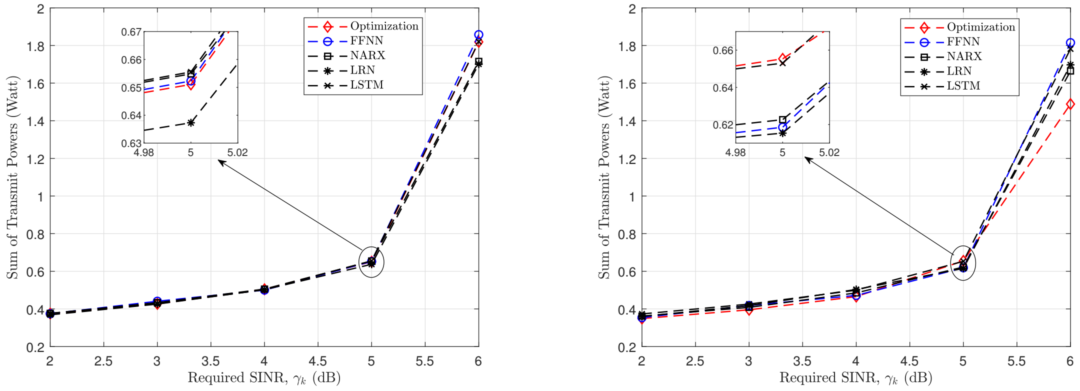
Figure 4.7: The computation time according to the number of samples when the number of the hidden layers is 2 and the hidden layer size is 20.

In addition, when considering the performance of the system based on the computation time, we also provide the computation time of the proposed schemes according to the number of samples when the number of the hidden layers is 2 and the hidden layer size is 20. Again, Fig. 4.7 also shows that the DL-based approaches still managed well with low computation time despite the significant increase in the number of samples. Meanwhile, the optimization algorithm provides an exponential computation time when the number of samples increases from 10000 to 50000.

The next simulation results provide more insight on the capability of the proposed DL-based approaches that can capture responses produced by the optimization algorithm in changing the required SINR and the required harvested energy.

4.4.2 Network Performance in Changing of Required SINR and Required Harvested Energy Values

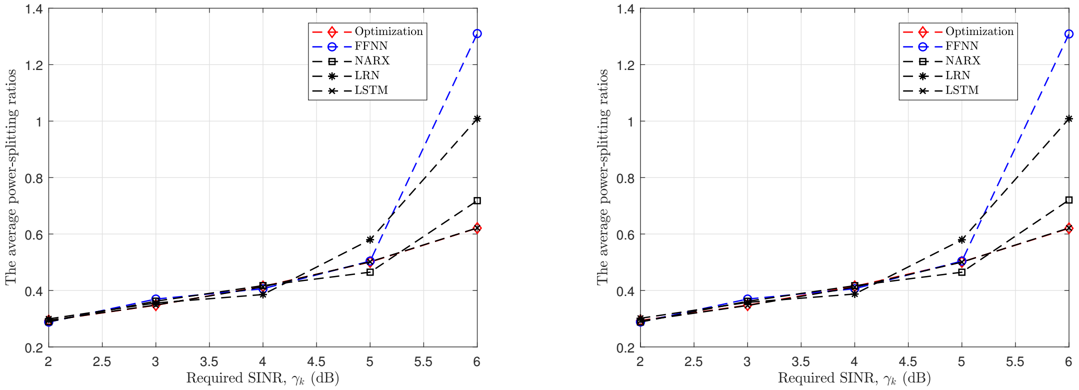
In this section, we provided the simulation results and discussions to verify the capability and efficiency of the proposed DL-based approaches in scenarios for changing the required SINR and required harvested energy. We employed two network performance metrics in the performance evaluation (transmit power and receiver PS ratio) with various network conditions, such as required SINR and required harvested energy. We also used two hidden layers, and 20 neurons for each hidden layer, as discussed in Section 4.4.1. Network condition parameters set for either changing the required SINR or the required harvested energy. In case of changing the required SINR, we fixed the required harvested energy value at −20 dBm and the required SINR is changed from 2 dB to 6 dB, as shown in Fig. 4.8 and Fig. 4.9 for the sum of the transmit powers and the average PS ratios, respectively. In the other cases, i.e, changing the required harvested energy, we fixed required SINR value at 2dB, and changed the required harvested energy from −20 dBm to −12 dBm, as shown in Fig. 4.10 and Fig. 4.11.



(a) Training stage

(b) Testing stage

Figure 4.8: Sum of transmit powers according to the required SINR when the required harvested energy is given by −20 dBm ($e_k = -20$ dBm).

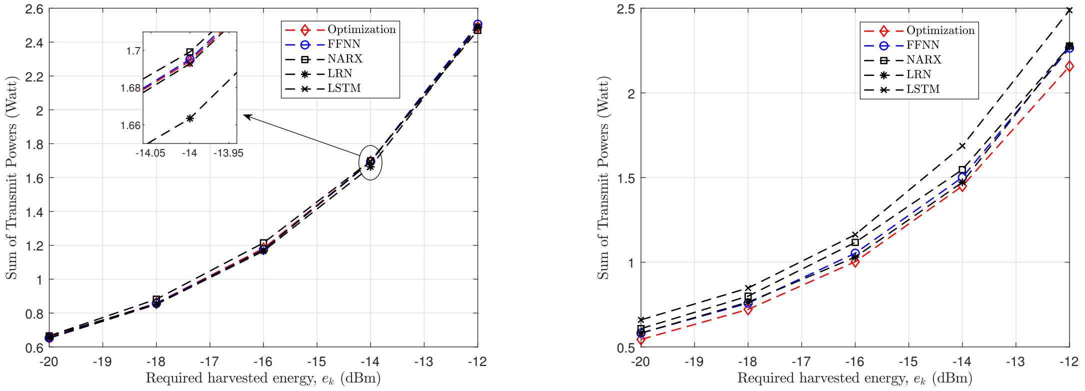


(a) Training stage

(b) Testing stage

Figure 4.9: Average power-splitting ratios according to the required SINR when the required harvested energy is given by -20 dBm ($e_k = -20$ dBm).

In Fig. 4.8 and Fig. 4.9, we show the sum of transmit powers and the average PS ratios, respectively, when changing the required SINR in the training and testing stages. Overall, the schemes mostly have an upward trend in both transmit power and PS ratio as the required SINR increases. This is because when the required SINR increases, the power of the transmitter must increase in order to guarantee the information services at the receivers. Moreover, the PS ratio increases to guarantee that the transmission is more effective so constraint (4.4b) can be guaranteed. When the required SINR is small, transmitters only use a small amount of power for transmission. This makes the fluctuation of transmit power among samples insignificant, and thus, the training and testing stages perform effectively, especially the sum of transmit powers (from 2 dB to 5 dB for the required SINR) and the average PS ratios (from 2 dB to 4 dB for the required SINR), as shown in Fig. 4.8 and Fig. 4.9, respectively. The result is that the DL-based approaches can capture a response similar to the response of the optimization algorithm in both training and testing stages. In other words, due to a relatively large fluctuation of transmit power that exists in the training data set at the high required SINR value ($\gamma_k = 6$ dB), the results of the training stage are less effective. Then, the testing stage will inevitably be less effective, as shown in Fig. 4.8b (at $\gamma_k = 6$ dB) and Fig. 4.9b (at $\gamma_k = 5$ dB and $\gamma_k = 6$ dB). Noteworthy is Fig. 4.9 at $\gamma_k = 6$ dB, where the results of the training and testing stages are not good, which make the PS ratio no longer significantly desirable ($\theta_k \notin (0, 1)$) in the FFNN and LRN approaches when a relatively large fluctuation of transmit power exists in the considered data. Unfortunately, the training process of traditional RNNs is influenced by an issue in backpropagation through time, called the exploding gradient (which can cause learning to diverge) or by the vanishing gradient, where learning either becomes very slow or stops working altogether (which prevents complete learning of the time series). One solution is to utilize LSTM networks. Therefore, in the comparison of simulation results in Fig. 4.8 and Fig. 4.9, LSTM has superior performance compared to the other DL-based approaches (in particular, in the average PS ratio for both training and testing stages as shown in Fig. 4.9). Moreover, we utilize relative error, which is a measure of the uncertainty of measurement to evaluate the variations between the results provided by the DL-based approaches and the optimization algorithm. For example, in Fig. 4.8a at required SINR is 5dB, relative errors of FFNN, NARX, LRN, and LSTM compared to optimization algorithm are 0.2%, 0.58%, 2.1%, and 0.69%, respectively. In Figure 4.8b at required SINR is 5dB, relative errors of FFNN, NARX, LRN, and LSTM compared to optimization algorithm are 5.6%, 5%, 6.1%, and 0.35%, respectively. Simulation results in Fig. 4.8a,b also show that LSTM provides the best performance in term of transmit power.

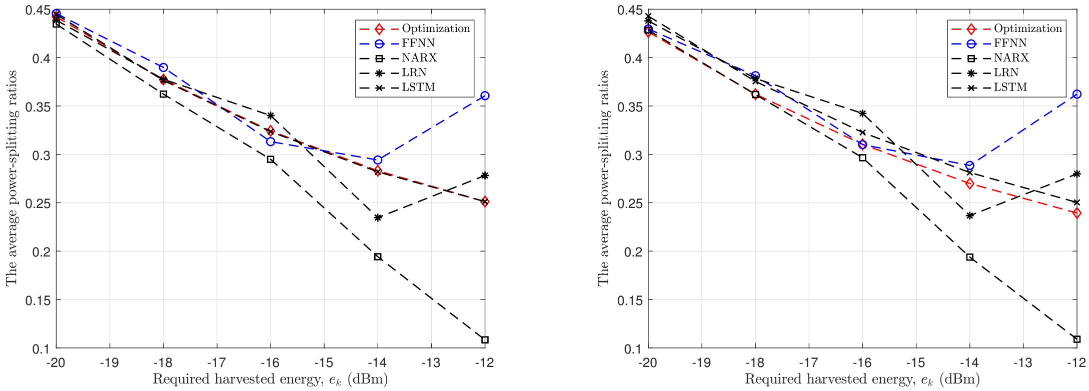


(a) Training stage

(b) Testing stage

Figure 4.10: Sum of transmit powers according to the required harvested energy when the required SINR is given by 2dB ($\gamma_k = 2$ dB).

Fig. 4.10 shows the sum of transmit powers while changing the required harvested energy in the training and testing stages. In Fig. 4.10, we can see that the change of required harvested energy threshold of receiver will affect the minimum transmit power of transmitter.



(a) Training stage

(b) Testing stage

Figure 4.11: Average power-splitting ratios according to the required harvested energy when the required SINR is given by 2dB ($\gamma_k = 2$ dB).

The higher the required harvested energy, the more power the transmitter needs to transmit, which makes the receivers avoid to interrupt the communication due to the lack of power. In Fig. 4.10a, the DL-based approaches can capture responses similar to the response of the optimization algorithm in the training stage with relatively large samples. However, with a smaller number of samples in the testing stage and the relatively large fluctuation of transmit powers in the testing data, the DL-based approaches cannot effectively capture the response of optimization algorithm as shown in Fig. 4.10b. We also calculate relative errors of FFNN, NARX, LRN, and LSTM compared to optimization algorithm in Figure 4.10a at the required harvested energy $e_k = -14$ dBm, which are 0.05%, 0.3%, 1.83%, and 0.06%, respectively. In this case, LSTM also provides low relative error in most of DL-based approaches. Fig. 4.11 shows the average PS ratios according to the required harvested energy in the training and testing stages. Overall, the schemes mostly have a downward trend in the PS ratio as the required harvested energy increases. That is, the PS ratio should be decreased to guarantee that the energy harvested by EH is utilized more efficiently. In Fig. 4.11a, FFNN, LRN, and NARX cannot approximate the optimization algorithm accurately. Therefore, these DL-based approaches cannot capture a response like the response of the optimization algorithm in testing stage, as shown in Fig. 4.11b. Although LSTM cannot capture a response as good as the response of the optimization algorithm in Fig. 4.10b, it gives the best performance in Fig. 4.11b, where it can capture a response similar to the response of the optimization algorithm.

4.5 Conclusion

In this work, we investigated the transmit power optimization problem in SISO interference channels with SWIPT systems where a PS scheme was used at each receiver. The transmit power optimization problems subject to the required SINR and required harvested energy were investigated by optimizing transmit power and the PS ratio. After that, we exploited the ability of DL to improve the computing time in comparison with the traditional optimization algorithm. The various approaches based on DL were proposed such as FFNN, LRN, NARX, and the LSTM network. The performance of the proposed approaches was evaluated and compared to that of the traditional optimization algorithm. From experimental results, LSTM network provided the best balance in the tradeoff between solution quality (i.e., MSE) and solution efficiency (i.e., computation time) in the DL-based approaches. Overall, the experimental results showed that DL models can forecast output of the optimization problem effectively without prior knowledge about the system's state. Most of all, DL-based approach provided low computation time, compared to the traditional optimization algorithm, which is very useful for real-time resource allocation processing.

From the simulation results, the FFNN is relatively simple to implement, so it has the lowest computation time. However, the FFNN provides inaccurate estimation of both transmit power and PS ratio. Due to the lack of control over the learning process, the FFNN further may lead to overfitting or underfitting to the training set, which makes predictions for new data be inaccurate. Traditional RNNs such as LRN and NARX have a "memory" which captures information about what has been calculated. In some cases of simulation results, traditional RNNs may improve network performance compared to the FFNN but in return longer computation time is required. However, traditional RNNs are influenced by the gradient vanishing and exploding problems, which makes the training of traditional RNN difficult, in two ways: (1) the RNN cannot process very long sequences if Hyperbolic tangent function is used as its activation function, (2) the RNN is very unstable if Rectified Linear Unit (ReLU) is used as its activation function. Besides, it cannot be stacked into very deep models. This is mostly due to the saturated activation function used in RNN models which makes the gradient decay over layers. LSTM network resolves these issues via Cell state (memory for LSTM) and Gates. In this chapter, simulation results also show that the LSTM network can provide the best balance in the tradeoff between solution quality (i.e., MSE) and solution efficiency (i.e., computation time) compared to the other DL-based approaches.

Chapter 5

A Secure-transmission Maximization Scheme for SWIPT Systems Assisted by an Intelligent Reflecting Surface and Deep Learning

5.1 Introduction

In recent years, wireless communication technologies have developed dramatically. The demand for quality of service (QoS) has also increased because of the rapid increase in the number of users, resulting in a scarcity of spectrum resources [113]. In addition, power consumption is constantly increasing due to expanding network infrastructure such as transmission lines, terminal equipment, and base stations (BSs). Therefore, it is becoming increasingly important to save energy. Efficient energy management helps to overcome the bottleneck of wireless network applications operating under battery and energy constraints. It not only helps to reduce a device's dependence on battery power and power consumption, but also provides a continuous power source for the long-term operation of devices on the network. As a result, the SWIPT transmission technique was developed to fulfill these requirements [4, 5, 71, 72]. In the SWIPT system, the received signal can be used for EH and

ID. In addition, to simultaneously perform power transfer and information transmission in a SWIPT system, two practical structures are used: PS and TS [4].

On the other hand, a SWIPT system also suffers negative effects, such as channel attenuation and interference signals from wireless transmission environments. Besides that, tall objects like trees, traffic signs, or buildings can block the communication link between the transmitter and the receivers in WCNs. All of them reduce the quality of the communication link and weaken the information and energy received. Fortunately, with the breakthrough developments in meta-materials in recent years, the intelligent reflecting surface (IRS) was developed and is considered an effective solution to overcome these negative effects [114, 115]. In addition, several variants of the IRS have been developed, such as the large intelligent surface (LIS) [116, 117], the large intelligent metasurface (LIM) [118], and the reconfigurable intelligent surface [119, 120]. An IRS includes an array of low-cost and passive reflecting elements. Each reflecting element is able to change the frequency, phase, amplitude, or even polarization of an incident signal [114, 115]. An IRS is introduced to generate an additional reflected link. Along with the signals directly received via direct communication links, an additional reflected signal can be added to suppress the channel interference of undesired receivers and improve the received signal for desired receivers. The IRS is more energy- and cost-efficient than a conventional relay system. This is because in a relay system, transmitting and receiving signals are done with the active RF signal. Meanwhile, for the IRS, the incident signal is reflected by reconfiguring the IRS's phase shifts without RF chains. So, the beamforming design in an IRS is classified in a nearly passive manner. The IRS also has quite low power consumption due to its lightweight and compact size, and thus, it is easily installed in the indoor environment (e.g., on ceilings and walls) and in outdoor environments (e.g., on road signs, moving trains, building facades, etc.).

Currently, we are living in an era of information and data explosion where sharing and exchanging information between devices takes place every day and hour. Personal data and private communications easily become targets of security threats such as eavesdroppers (Eave's) [121]. Thus, a private conversation or communication in a SWIPT advanced networking system, even in combination with an IRS, may be secretly or stealthily overheard. Therefore, the secure transmission problem in IRS-enabled SWIPT systems must be considered more and more important.

From the above surveys, for the purposes of efficient energy management, secure transmission and signal enhancement from the IRS technique, we investigate the secure transmission IRS-enabled SWIPT system which is one of the current research topics of interest.

5.2 Related works

There has already been a lot of work investigating the system secrecy rate optimization problem with many modern techniques applied, such as artificial noise (AN)-based anti-jamming, and multi-antenna beamforming [121–123], or even in the SWIPT system ifself [124, 125]. Liu *et al.* [124] studied the secure transmission optimization problem in SWIPT systems with multiple energy receivers (ERs) and an information receiver (IR). They aimed to optimize the ERs' weighted sum energy and the IR's secrecy rate. The authors in [125] maximized the system sum secrecy rate by satisfying the constraints on the ER's minimum harvested energy and the IR's minimum data rate. The secure transmission problem was addressed in a SWIPT-enabled non-orthogonal multiple access (NOMA) system that consisted of multiple IRs, multiple ERs, and a BS. Studies on optimal secrecy rates have also been conducted in WCNs with the help of IRSs [126, 127], where the authors considered an IRS-assisted wireless transmission system in which a single-antenna eavesdropper attempts to listen to communications. The secrecy rate is maximized by optimizing the IRS's reflect beamforming and the transmitter's beamforming. Both systems used an alternating optimization (AO) algorithm for solving optimization problems. Simulation results showed a significant improvement in terms of the secrecy communication rate from the proposed scheme compared to a scheme not using an IRS.

Furthermore, there have been many studies on the secure transmission of IRS-assisted SWIPT systems. However, the secrecy rate optimization problem was not considered as the main optimization problem [128–131]. More specifically, the authors in [128–130] aim to optimize transmit beamforming while ensuring the constraints of the QoS and harvested energy. Niu *et al.* [131] maximized the minimum robust information rate among the legitimate IRs while the ERs are considered as potential Eave's. In addition, the IRS-assisted SWIPT system was considered in [132, 133] to optimize secrecy rate. However, the PS factor was not jointly optimized in the secure transmission of IRS-assisted SWIPT system, which is an important factor that can prolong the uptime and improve the energy efficiency of devices.

Summary, in all of the aforementioned work, the secure transmission optimization problem was mostly considered in following system models: the SWIPT system without IRS,

Table 5.1: The comparison of existing works related to IRS and SWIPT systems.

Group	**System model [related work]**	**Objective**
The SWIPT system without IRS	SWIPT [124]	to optimize the ER's weighted sum energy and IR's secrecy rate
	SWIPT + NOMA [125]	to maximize the system sum secrecy rate
The conventional IRS system	IRS + WCN [126, 127]	to optimize secrecy rate
The IRS-assisted SWIPT system (without secrecy rate optimization)	IRS + SWIPT [128]	to minimize a total transmit power
	IRS + SWIPT + NOMA + PS [129]	to minimize BS transmit power
	IRS + SWIPT + PS [130]	to minimize a total transmit power
	IRS + SWIPT [131]	to maximize the minimum robust information rate among the IRs
The IRS-assisted SWIPT system (secrecy rate optimization + no PS scheme)	IRS + SWIPT + no PS [132, 133]	to maximize secrecy rate
	IRS + SWIPT + non-linear harvested energy model + no PS [134]	to optimize secrecy rate
Our work	IRS + SWIPT + PS	to optimize secrecy rate and to improve computation performance using DL

the conventional IRS-assisted WCN system, the IRS-assisted SWIPT system with no secrecy rate optimization, and the IRS-assisted SWIPT system with secrecy rate optimization without considering PS scheme. Most recently, the secure transmission optimization problem was studied in an IRS-assisted SWIPT system where separate receivers are IRs and ERs [134]. At the ERs, the harvested energy was formulated by a practical non-linear model. In addition, the secrecy rate was maximized while constraints on EH and transmit power for the ERs and the BS being satisfied, by optimizing the AN covariance, the BS's transmit beamforming, and the IRS's reflective beamforming. The AO algorithm was also implemented to solve the target problem. However, it is noteworthy that our work is different from [134], although the secure transmission issue is also considered in the IRS-assisted SWIPT system. In this work, we consider the unified UE with a PS scheme where the secrecy rate should be maximized by additionally considering the PS factor at the UE. Furthermore, in our work, the computational efficiency of the optimization algorithm is also studied in comparison with the proposed DL-based approach, which the previous works did not take into account. Table 5.1 compares existing works related to IRS and SWIPT systems.

Although the optimization algorithm-based approach is a very powerful approach for solving most optimization problems including convex and non-convex problems, it still faces many challenges when deployed in many applications with low computation time requirements. This disadvantage comes from the implementation of optimization algorithms, which are based on iterations and complex mathematical transitions from non-convex problems to convex problems. Fortunately, the DL technique can effectively overcome these issues. DL technology has shown high efficiency when applied in WCNs [92]. Sun *et al.* [112] investigated the WMMSE discussed in [83], and the interference was approximated by using a DNN. Results showed that the WMMSE problem can be well-approximated with low computation time through a DNN model.

In this chapter, to take advantage of the IRS and SWIPT system, we investigate an IRS-assisted SWIPT systems in which the IRS is deployed to improve the security of the communication link between a single-antenna transmitter and a single-antenna UE despite eavesdropping by a single-antenna Eave'. We not only study the secure transmission optimization problem in the IRS-assisted SWIPT system with a PS scheme in the UE, but we also consider a neural network for achieving computational efficiency. The optimization problem of secure transmission is difficult to solve when it has non-convex form. Fortunately, the optimization problems with non-convex form can be effectively solved using the feasible point pursuit–successive convex approximation (FPP–SCA) algorithm [135] and the AO method [126, 127, 134]. The FPP–SCA algorithm executes the non-convex functions (non-convex constraints, or even non-convex objective functions) with upper convex functions at each iteration. Specifically, the concave terms are approximated around a feasible point by a convex function, and the optimal solution of the convex problem in the current iteration will be served for the next iteration as the feasible point. On the other hand, the AO method optimizes one or more variables by fixing remaining variables in an alternating manner. Regarding the DL-based approach, training and running stages are required. After the optimization algorithm reaches feasible solutions, the optimal output along with the corresponding input will be used as the training data for DNN model. If the DNN is well-trained (i.e., the trained network can provide predictive outputs almost identical to the feasible solutions of the optimization algorithm), then, the trained DNN can be applied to estimate optimal output in the running stage with lower computation time.

In a nutshell, this chapter's main contributions are as follows.

- We consider an IRS-assisted SWIPT system where a signal is transmited to the UE while an Eave' tries to listen to the transmitter–UE communication. By deploying an IRS in the system, network security can be enhanced, and eavesdropping can be reduced. Furthermore, the UE is equipped with a PS scheme that makes the UE get both signal and harvested energy simultaneously. We formulate the secure transmission problem of an IRS-assisted SWIPT system with a PS scheme to maximize the system secrecy rate by finding the optimal solutions for the transmitter's power, the UE's PS factor, and the IRS's phase shifts matrix.

- We propose an AO-based scheme for solving the optimization problem where FPP, SCA, and penalty methods are used to solve the optimization problem.

- A DL-based approach is considered to improve computational performance. Specifically, 5 types of data and DNN structures are proposed.

Notations: Matrices and vectors are denoted by boldface capital and lower-case letters, respectively, while $(\cdot)^H$ and $(\cdot)^T$ represent the Hermitian and the transpose operations, respectively. The scalar's absolute value is denoted by $|\cdot|$. The diagonal matrix is represented by $diag\left\{\cdot\right\}$ where the elements of the input vector are diagonal. $\mathbb{C}^{m\times n}$ represents a complex matrix with an $m \times n$ space. $\mathcal{CN}\left(0, \sigma^2\right)$ denotes the random variable distribution with zero mean and variance σ^2 of a circularly symmetric complex Gaussian (CSCG), and '$\sim$' implies *distributed as*. The symbols $\mathbb{E}\{\cdot\}$ and $\mathrm{Tr}(\cdot)$ represent expectation and trace operations. Defining $\boldsymbol{Q} \succeq 0$ means $\boldsymbol{Q}$ is a positive semi-definite (PSD) matrix. The terms $\mathrm{Im}\,(a)$ and $\mathrm{Re}\,(a)$ represent the imaginary part and the real part of complex number a. Table 5.2 lists other notations used in this chapter.

The subsequent sections of this chapter are organized as follows. Section 5.3 presents the formulation of the problem with the system model, the proposed AO-based scheme and the proposed DL-based approach. Analysis and discussion of the simulation results are in Section 5.4. Finally, Section 5.5 presents the conclusion.

Table 5.2: The notation list.

Symbol	**Description**
M	the number of reflecting elements
$\boldsymbol{\Phi}$	IRS's diagonal phase shifts matrix
ϕ_m	phase shift of the m–th reflecting element
$\mathbf{w}$ and z	horizontal coordinates and altitude, respectively
$h_{TU}, h_{TE}, \boldsymbol{h}_{TI},\ \boldsymbol{h}_{IU}, \boldsymbol{h}_{IE}$	channel gains of the communication links
$\alpha_{TU}, \alpha_{TE}, \alpha_{TI}, \alpha_{IU}, \alpha_{IE}$	path loss exponents of the communication links
$d_{TU}, d_{TE}, d_{TI}, d_{IU}, d_{IE}$	distances of the communication links
ρ_l	path loss at reference distance $D_0 = 1$ m
$\beta_{TI}, \beta_{IU}, \beta_{IE}$	Rician factors of the communication links
λ_c and Δ	carrier wavelength and antenna separation, respectively
ψ	cosine of the angle
P	transmission power at the transmitter
θ	power-splitting (PS) factor
R_{sec}	secrecy rate
P_{max}	required maximum transmitter power
e	required minimum harvested energy
$\lambda_{max}\left(\boldsymbol{Q}\right)$	maximum eigenvalue of $\boldsymbol{Q}$
$\boldsymbol{w}_{max}$	eigenvector corresponding to maximum eigenvalue $\lambda_{max}\left(\boldsymbol{Q}\right)$

5.3 Formulation of the Problem

5.3.1 Channel model

In this work, an IRS-assisted SWIPT system is considered, consisting of a transmitter, a UE, an Eave', and an IRS (as shown in Fig. 5.1). The UE, transmitter, and the Eave' utilizes a single omni-directional antenna, respectively, while M reflecting elements are used in a uniform linear array (ULA), which is the IRS indexed by $\mathcal{M} \triangleq \{1, \ldots, M\}$. The IRS is connected to a smart controller, which can configure the IRS phase shifts in real-time manner for desired signal propagation [120, 136]. The UE is equipped with a PS scheme. The IRS is placed parallel to the x-axis and is located in the $x - z$ plane. Let $\mathbf{\Phi} = \text{diag}\left\{\phi_1, \phi_2, \ldots, \phi_M\right\} \in \mathbb{C}^{M \times M}$ be the IRS's diagonal phase shifts matrix, where $\phi_m = \beta_m e^{j\varphi_m}$ is the phase shift of the m–th reflecting element with $\varphi_m \in [0, 2\pi)$ and $\beta_m \in [0, 1]\,, \forall m \in \mathcal{M}$. The phase shifts $\{\phi_m\}$ can be controlled continuously and β_m is the amplitude reflection coefficient of the m-th reflecting element. In practice, when designing elements of the IRS, the amplitude reflection coefficient is often set to 1 to achieve maximum signal reflection such that we have $\beta_m = 1, \forall m$. In addition, we assume that the center point of the IRS is the reference point, where the horizontal coordinates and altitude are indicated by $\mathbf{w}_I = [x_I, y_I]^T$ and z_I, respectively. Therefore, the distance of the communication link from a particular user node to the IRS can be approximately equal to the distance from the corresponding user node to the reference point of the IRS. The horizontal coordinates of the transmitter, the UE, and the Eave' are denoted by $\mathbf{w}_T = [x_T, y_T]^T, \mathbf{w}_U = [x_U, y_U]^T$, and $\mathbf{w}_E = [x_E, y_E]^T$, respectively.

Because the location of Eave' is uncertain, the knowledge of the CSI between transmitter and Eave' is difficult to achieve. However, many methods and assumptions have been considered in recent studies to solve this problem. This knowledge may range from a complete lack of CSI (the approach based on studying the compound wiretap channel [137]) to partial CSI (optimizing the AN transmit covariance [138] or relaxing the orthogonality constraint [139]) and statistical CSI (meeting a target performance criterion in terms of SNR or rate at the receiver based on allocating enough power [121]) or even the CSI uncertainty (adopting a deterministic model [140–142]). In addition, there are some methods to identify the presence of an Eave' such as detection-theoretic methods based on its local oscillator leakage power and mutual communication between the legitimate nodes based on realizations of a constructed random variable [143]. Moreover, it is reasonable to assume that the CSIs

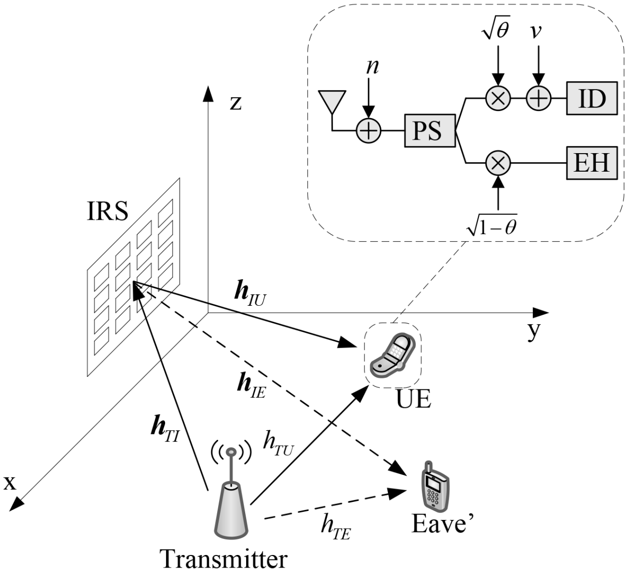


Figure 5.1: The IRS-assisted SWIPT system with a power-splitting (PS) scheme in UE.

of links related to Eave' can be known when the Eave' is considered as an active user but untrusted by the legitimate user [121]. Besides, several channel estimation techniques for IRS-assisted systems have been proposed recently such as those mentioned in [118, 144]. Therefore, to characterize the performance limit of the secure transmission IRS-assisted SWIPT system, the CSIs of the channels involved are assumed to be either completely known at the BS/IRS or achievable based on existing channel estimation techniques. In general, for the sake of simplicity in our scenario, the CSIs of the channels involved are modeled as the Rayleigh and Rician fading channels as follows.

Let $h_{TU} \in \mathbb{C}^{1\times 1}$ and $h_{TE} \in \mathbb{C}^{1\times 1}$, respectively, denote the channel gain of transmitter–UE (T–U) and transmitter–Eave' (T–E) links. We assume the channel gain of the T–U and T–E links model a the Rayleigh fading channel, as follows:

$$h_{TU} = \sqrt{\rho_l d_{TU}^{-\alpha_{TU}}}\tilde{h}_{TU}, \tag{5.1}$$

$$h_{TE} = \sqrt{\rho_l d_{TE}^{-\alpha_{TE}}}\tilde{h}_{TE}, \tag{5.2}$$

where $\tilde{h}_{TU}$ and $\tilde{h}_{TE}$ denote the CSCG random variable distribution, d_{TU} and d_{TE} denote the distances of the corresponding communication links, calculated by $d_{TU} = \sqrt{\|\mathbf{w}_T - \mathbf{w}_U\|^2}$ and $d_{TE} = \sqrt{\|\mathbf{w}_T - \mathbf{w}_E\|^2}$, respectively, α denotes the path loss exponent, and ρ_l denotes the path loss at reference distance $D_0 = 1$ m [145].

In fact, the IRS can be installed on the facade of the building so the links from the transmitter to the UE, and from the IRS to the UE and to the Eave' might not be blocked by obstructions like trees or traffic signs. As a result, there is a line-of-sight (LoS) component to these channels. Hence, with the addition of the LoS path, the channel gain of the transmitter–IRS (T–I), the IRS–UE (I–U), and the IRS–Eave' (I–E) links can model a Rician fading channel. Let $cl = \{TI, IU, IE\}$ denote the communication links of the T–I, I–U, and I–E links. Then, the channel gain of the related communication links, $\boldsymbol{h}_{cl} \in \mathbb{C}^{M\times 1}$, can be expressed as follows:

$$\boldsymbol{h}_{cl} = \sqrt{\rho_l d_{cl}^{-\alpha_{cl}}} \left(\sqrt{\frac{\beta_{cl}}{1+\beta_{cl}}} \boldsymbol{h}_{cl}^{LoS} + \sqrt{\frac{1}{1+\beta_{cl}}} \boldsymbol{h}_{cl}^{NLoS} \right), \tag{5.3}$$

where d_{cl}, α_{cl}, and β_{cl} represent the distances, the path loss exponents, and the Rician factors of the related communication links, cl. The distances of the related communication links, $d_{cl} = \{d_{TI}, d_{IU}, d_{IE}\}$, are calculated as follows: $d_{TI} = \sqrt{z_I^2 + \|\mathbf{w}_T - \mathbf{w}_I\|^2}$, $d_{IU} = \sqrt{z_I^2 + \|\mathbf{w}_U - \mathbf{w}_I\|^2}$, and $d_{IE} = \sqrt{z_I^2 + \|\mathbf{w}_E - \mathbf{w}_I\|^2}$. The non-LoS (NLoS) and LoS components of a communication link are denoted by $\boldsymbol{h}_{cl}^{NLoS}$ and $\boldsymbol{h}_{cl}^{LoS}$, respectively. The NLoS component of the communication link, $\boldsymbol{h}_{cl}^{NLoS} \in \mathbb{C}^{M\times 1}$, follows a CSCG random variable distribution, wheares the LoS component, $\boldsymbol{h}_{cl}^{LoS} \in \mathbb{C}^{M\times 1}$, is a ULA of M elements [146] and is given as:

$$\boldsymbol{h}_{cl}^{LoS} = \left[1, e^{-j\frac{2\pi}{\lambda_c}\Delta\psi_{cl}}, \ldots, e^{-j\frac{2\pi}{\lambda_c}(M-1)\Delta\psi_{cl}}\right], \tag{5.4}$$

where λ_c and Δ represent the carrier wavelength and the antenna separation, respectively; $\psi_{cl} = \{\psi_{TI}, \psi_{IU}, \psi_{IE}\}$ denotes the cosine of the angle of the related communication links, in which $\psi_{TI} = \frac{x_I - x_T}{d_{TI}}$ denotes the cosine of the angle of arrival (AoA) for the propagation path from the transmitter to the IRS, while $\psi_{IU} = \frac{x_U - x_I}{d_{IU}}$ and $\psi_{IE} = \frac{x_E - x_I}{d_{IE}}$ denote the cosine of the angle of departure (AoD) of the propagation paths from the IRS to the UE and to the Eave', respectively.

5.3.2 Communication model

The transmitter sends signal $x_t = \sqrt{P}s$, where s denotes the information-bearing symbol, which is a CSCG distribution. P denotes the transmitter power, and $\mathbb{E}\left\{|s|^2\right\} = 1$. In this work, the IRS is assumed to be able to impose an additional time delay on the incident signals, which not only helps the coherent superposition of multiple copies of the desired signals but also guarantees their synchronization in time. Specifically, one of the possible approaches is the delay adjustable elements [147] cascaded with the existing phase adjustable elements [114]. In addition, to ensure that the incident signals are reflected independently by all IRS elements, the reflected signal-coupling among neighboring IRS elements is assumed that does not exist. Moreover, due to the severe path loss, we only consider signals which are reflected by the IRS first time [145, 148] by ignoring signals which are reflected by the IRS two or more times. The received signals at the UE and the Eave' are defined as follows:

$$y_U = h_U \sqrt{P} s + n_U, \tag{5.5}$$

$$y_E = h_E \sqrt{P} s + n_E, \tag{5.6}$$

where $h_U = \boldsymbol{h}_{IU}^H \boldsymbol{\Phi} \boldsymbol{h}_{TI} + h_{TU}$ and $h_E = \boldsymbol{h}_{IE}^H \boldsymbol{\Phi} \boldsymbol{h}_{TI} + h_{TE}$, with $n_U \sim \mathcal{CN}\left(0, \sigma_U^2\right)$ and $n_E \sim \mathcal{CN}\left(0, \sigma_E^2\right)$ denoting noise from the antenna at the UE and the Eave', respectively. By using the PS scheme, the UE is able to execute EH and ID simultaneously. Regarding the PS structure, the received signal can be divided into ID and EH streams with PS factors θ and $(1-\theta)$, respectively, where $\theta \in (0,1)$. The ID process is only executed on the ID stream at the UE, and thus, the signal-to-noise ratio (SNR) at the UE and the Eave' can be obtained. Accordingly, the achievable rates at the UE and the Eave' are defined as follows:

$$R_U = \log_2\left(1 + \frac{\theta |h_U|^2 P}{\theta \sigma_U^2 + \delta_U^2}\right), \tag{5.7}$$

$$R_E = \log_2\left(1 + \frac{|h_E|^2 P}{\sigma_E^2}\right), \tag{5.8}$$

where $v \sim \mathcal{CN}\left(0, \delta_U^2\right)$ is the noise of the circuit on the ID stream at the UE shown in Fig. 5.1. Regarding the EH stream, the EH process is executed, and thus, the harvested energy at the UE is determined as:

$$EH_U = \mu\left(1-\theta\right)\left(|h_U|^2 P + \sigma_U^2\right), \tag{5.9}$$

where $\mu \in (0,1]$ and denotes the efficiency of the EH process on the EH stream at the UE. In this work, for simplicity in computation, the UE is assumed to harvest all the energy from the received signal, and thus, μ is fixed at 1 ($\mu = 1$) for the remainder of this chapter.

5.3.3 Objective Problem

For secure transmission, the success of the user is maximized, whereas the success of the Eave' needs to be minimized. To estimate this performance metric the *secrecy rate* is often used, defined as the variation between the achievable rates of the user and the Eave' [121]. Therefore, the secrecy rate at the UE in bits/second/Hertz (bps/Hz) is given as follows:

$$R_{sec}\left(P,\theta,\mathbf{\Phi}\right) = \left(R_U - R_E\right)^{+}, \tag{5.10}$$

where the function $(x)^{+} = \max\left(x,0\right)$.

In this work, we aim to maximize the system secrecy rate by optimizing the received PS factor, θ, the transmitter power, P, and the phase shifts matrix, $\mathbf{\Phi}$, subject to constraints on the required harvested energy and power. Then, the secrecy rate optimization problem is formulated as:

$$\begin{aligned}
\max_{P,\theta,\mathbf{\Phi}} \quad & R_{sec}\left(P,\theta,\mathbf{\Phi}\right) && \text{(5.11a)}\\
\text{s.t.:} \quad & (1-\theta)\left(\left|h_U\right|^2 P + \sigma_U^2\right) \geq e, && \text{(5.11b)}\\
& P \leq P_{max}, && \text{(5.11c)}\\
& 0 < \theta < 1, && \text{(5.11d)}\\
& \left|\phi_m\right| = 1, \quad \forall m \in \mathcal{M}, && \text{(5.11e)}
\end{aligned}$$

where P_{max} denotes the required maximum transmitter power, and e represents the required minimum harvested energy.

5.3.4 The proposed AO-based scheme for the secure transmission problem

In this section, we propose an AO-based algorithm for solving problem (5.11) which provides optimal value of P,θ and $\mathbf{\Phi}$ in an alternating manner. Since the AO method optimizes one or more variables by fixing remaining variables in an alternating manner, in the proposed scheme, the optimization of P,θ with a fixed $\mathbf{\Phi}$ is found by the FPP–SCA method, while the optimization of $\mathbf{\Phi}$ with a given P,θ is found by FPP–SCA and a penalty method.

5.3.4.1 Finding P, θ with a given $\mathbf{\Phi}$

Since $\mathbf{\Phi}$ is fixed, the constraint (5.11e) is satisfied. Then, we remove the logarithm function and add two variables u, v where $u, v \geq 0$. Specifically, u^2 is used for the numerator and v is used for the denominator of the problem. Then, we can get the inequality $\left(1 + \frac{\theta|h_U|^2 P}{\theta\sigma_U^2 + \delta_U^2}\right) \Big/ \left(1 + \frac{|h_E|^2 P}{\sigma_E^2}\right) \geq u^2/v$ which is always guaranteed. Therefore, problem (5.11) can be changed to:

$$\max_{P,\theta,u,v} \quad \frac{u^2}{v} \tag{5.12a}$$

$$\text{s.t.:} \quad 1 + \frac{|h_U|^2 P}{\sigma_U^2 + \frac{\delta_U^2}{\theta}} \geq u^2, \tag{5.12b}$$

$$1 + \frac{|h_E|^2 P}{\sigma_E^2} \leq v, \tag{5.12c}$$

$$(1-\theta)\left(|h_U|^2 P + \sigma_U^2\right) \geq e, \tag{5.12d}$$

$$P \leq P_{max}, \tag{5.12e}$$

$$0 < \theta < 1, \tag{5.12f}$$

$$u, v, P \geq 0. \tag{5.12g}$$

Then, problem (5.12) can be changed to:

$$\min_{P,\theta,u,v} \quad -\frac{u^2}{v} \tag{5.13a}$$

$$\text{s.t.:} \quad 0 \geq \sigma_U^2 u^2 + \frac{\delta_U^2 u^2}{\theta} - \frac{\delta_U^2}{\theta} - |h_U|^2 P - \sigma_U^2, \tag{5.13b}$$

$$0 \geq 1 + \frac{|h_E|^2}{\sigma_E^2} P - v, \tag{5.13c}$$

$$0 \geq \frac{1}{1-\theta} e - |h_U|^2 P - \sigma_U^2, \tag{5.13d}$$

$$0 \geq P - P_{max}, \tag{5.13e}$$

$$0 < \theta < 1, \tag{5.13f}$$

$$u, v, P \geq 0. \tag{5.13g}$$

Due to the non-convex property of $\left(-\frac{u^2}{v}\right)$ and $\left(-\frac{1}{\theta}\right)$ in the objective function (5.13a) and under constraint (5.13b), respectively, problem (5.13) is non-convex. Therefore, we need to perform first-order Taylor approximation as follows:

$$\frac{u^2}{v} \geq \frac{{u^{(n)}}^2}{v^{(n)}} + \left[\begin{array}{cc} \frac{2u^{(n)}}{v^{(n)}} & \frac{-{u^{(n)}}^2}{{v^{(n)}}^2} \end{array} \right] \left(\left[\begin{array}{c} u \\ v \end{array} \right] - \left[\begin{array}{c} u^{(n)} \\ v^{(n)} \end{array} \right] \right) = \frac{2u^{(n)}u}{v^{(n)}} - \frac{{u^{(n)}}^2 v}{{v^{(n)}}^2}, \tag{5.14}$$

$$\frac{1}{\theta} \geq \frac{1}{\theta^{(n)}} - \frac{1}{{\theta^{(n)}}^2}\left(\theta - \theta^{(n)}\right) = \frac{2}{\theta^{(n)}} - \frac{\theta}{{\theta^{(n)}}^2}. \tag{5.15}$$

After that, non-convex problem (5.13) is reformulated into an approximated convex problem with the n–th sub-problem using the FPP–SCA method. From the FPP–SCA method [135], slack variables s_1, s_2, s_3, s_4 are also added into problem (5.13) to generate a feasible point. By replacing (5.14) and (5.15), and adding the slack variables into problem (5.13), non-convex problem (5.13) is converted into a convex problem as follows:

$$\min_{\substack{P,\theta,u,v,\\ s_1,s_2,s_3,s_4}} \quad -\frac{2u^{(n)}u}{v^{(n)}} + \frac{{u^{(n)}}^2 v}{{v^{(n)}}^2} + \lambda\left(s_1 + s_2 + s_3 + s_4\right) \tag{5.16a}$$

$$\text{s.t.:} \quad 0 \geq A - \delta_U^2\left(\frac{2}{\theta^{(n)}} - \frac{\theta}{{\theta^{(n)}}^2}\right) - s_1, \tag{5.16b}$$

$$0 \geq 1 + \frac{|h_E|^2}{\sigma_E^2}P - v - s_2, \tag{5.16c}$$

$$0 \geq \frac{1}{(1-\theta)}e - |h_U|^2 P - \sigma_U^2 - s_3, \tag{5.16d}$$

$$0 \geq P - P_{max} - s_4, \tag{5.16e}$$

$$0 < \theta < 1, \tag{5.16f}$$

$$u, v, P, s_1, s_2, s_3, s_4 \geq 0, \tag{5.16g}$$

where $A = \sigma_U^2 u^2 + \frac{\delta_U^2 u^2}{\theta} - |h_U|^2 P - \sigma_U^2$ and λ is a trade-off factor between the slack term and the objective function. In this work, by using the convex Taylor underestimation of (5.14) and (5.15) and with the FPP–SCA method, problem (5.13) is transformed into solvable convex problem (5.16). Then, by using the interior-point method [149, 150] with a solver tool like Matlab's CVX [107], the convex optimization problem will be solved easily. Finally, the proposed FPP–SCA algorithm is presented in Algorithm 5.1.

Regarding Algorithm 5.1, because problem (5.11) is converted to convex sub-problem (5.16) in the n–th iteration , we need to determine initial feasible points $\left(P^{(0)}, \theta^{(0)}, u^{(0)}, v^{(0)}\right)$. We can set $P^{(0)} = P_{max}$ and $\theta^{(0)} = 0.5$ such that constraints (5.12e) and (5.12f) are satisfied, respectively. After that, $u^{(0)}$ and $v^{(0)}$ can be calculated such that constraints (5.12b) and (5.12c) are satisfied, respectively (step 4 in Algorithm 5.1).

In Algorithm 5.1, in each of the iterations and using Matlab's CVX to solve the convex problem, we then obtain the optimal solution. We assign $P^{(n)} \leftarrow P^{(n-1)^*}$, $\theta^{(n)} \leftarrow \theta^{(n-1)^*}$, and $u^{(n)} \leftarrow u^{(n-1)^*}$, $v^{(n)} \leftarrow v^{(n-1)^*}$, where $\left(P^{(n-1)^*}, \theta^{(n-1)^*}, u^{(n-1)^*}, v^{(n-1)^*}\right)$ is the optimal solution from the previous iteration. In other words, the $(n+1)$–th iteration reuses the optimal solution at the n–th iteration which is considered a feasible point. Therefore, the optimal solution u^* and v^* will be assigned for the next iterations of u and v, $u^{(n+1)}$ and $v^{(n+1)}$, respectively. This process will be repeated until convergence. Besides, Algorithm 5.1 converges when the initial point is feasible and converges to a stationary point. This is because the alternative functions $\left(-\frac{2u^{(n)}u}{v^{(n)}} + \frac{{u^{(n)}}^2 v}{{v^{(n)}}^2}\right)$ and $\left(\frac{2}{\theta^{(n)}} - \frac{\theta}{{\theta^{(n)}}^2}\right)$ satisfy the convergence conditions as mentioned in Section II-C [151]. Furthermore, the FPP–SCA approach yields a non-increasing cost sequence when considering additional slack variables which make the optimal values non-increasing after each iteration [135], and thus, Algorithm 5.1 is guaranteed to be converged.

Algorithm 5.1 The FPP–SCA algorithm obtaining transmit power (P) and PS factor (θ) by fixing phase shift matrix $\mathbf{\Phi}$

```
Input: Channel gain of the related communication links ($h_{TU}, h_{TE}, \boldsymbol{h}_{TI}, \boldsymbol{h}_{IU}, \boldsymbol{h}_{IE}$), fixed initial phase shifts matrix $\mathbf{\Phi}$, convergence conditions ($\varepsilon_1$, $\varepsilon_2$), a feasible point with an initial point $u^{(0)}, v^{(0)}, P^{(0)}, \theta^{(0)}$, $\lambda = 100$, required minimum harvested energy ($e$), required maximum transmit power ($P_{max}$), and $n = 0$.
Output: The optimal value: $P^*, \theta^* \Rightarrow R^*_{sec}$.
Calculate: $h_U = \boldsymbol{h}_{IU}^H \mathbf{\Phi} \boldsymbol{h}_{TI} + h_{TU}$ and $h_E = \boldsymbol{h}_{IE}^H \mathbf{\Phi} \boldsymbol{h}_{TI} + h_{TE}$
Calculate initial feasible point at $n = 0$: choose $P^{(0)} = P_{max}, \theta^{(0)} = 0.5$ such that they satisfy constraints (5.12e) and (5.12f), respectively. Then, calculate $u^{(0)}$ and $v^{(0)}$ such that $u^{(0)}$ and $v^{(0)}$ satisfy constraints (5.12b) and (5.12c), respectively, $u^{(0)} = \sqrt{1 + \frac{P^{(0)}|h_U|^2}{\sigma_U^2 + \frac{\delta_U^2}{\theta^{(0)}}}}$, $v^{(0)} = 1 + \frac{P^{(0)}|h_E|^2}{\sigma_E^2}$
repeat
    Solve problem (5.16) using Matlab CVX solver and calculate: $P^*, \theta^*, u^*, v^*, s_1^*, s_2^*, s_3^*, s_4^*$
    $n = n + 1$
    Calculate: $R^{(n-1)} = -\frac{2u^{(n-1)}u^*}{v^{(n-1)}} + \frac{{u^{(n-1)}}^2 v^*}{{v^{(n-1)}}^2} + \lambda\left(s_1^* + s_2^* + s_3^* + s_4^*\right)$
    Update: $P^{(n)} \leftarrow P^*$, $\theta^{(n)} \leftarrow \theta^*$, $u^{(n)} \leftarrow u^*$, $v^{(n)} \leftarrow v^*$
until $\frac{R^{(n)} - R^{(n-1)}}{R^{(n-1)}} \leq \varepsilon_1$ and $s_1 + s_2 + s_3 + s_4 \leq \varepsilon_2$
return $P^* \leftarrow P^{(n)}$, $\theta^* \leftarrow \theta^{(n)}$, calculate optimal secrecy rate $R^*_{sec}$ based on $P^*$ and $\theta^*$.
```

5.3.4.2 Finding $\mathbf{\Phi}$ with a given P, θ

By removing the logarithm function, performing some computational operations, and fixing P, θ, problem (5.11) with regard to (w.r.t.) $\mathbf{\Phi}$ becomes:

$$\max_{\mathbf{\Phi}} \quad \frac{\frac{1}{B}|h_U|^2+1}{\frac{1}{C}|h_E|^2+1} \tag{5.17a}$$

$$\text{s.t.:} \quad (1-\theta)\left(|h_U|^2 P+\sigma_U^2\right) \geq e, \tag{5.17b}$$

$$|\phi_m| = 1, \quad \forall m \in \mathcal{M}, \tag{5.17c}$$

where $B = \frac{\sigma_U^2+\delta_U^2/\theta}{P}$ and $C = \frac{\sigma_E^2}{P}$. Let $\boldsymbol{a}_1 = \mathrm{diag}\big(\boldsymbol{h}_{IU}^H\big)\boldsymbol{h}_{TI}$, $a_2 = h_{TU}$, $\boldsymbol{a}_3 = \mathrm{diag}\left(\boldsymbol{h}_{IE}^H\right)\boldsymbol{h}_{TI}$, $a_4 = h_{TE}$, $\boldsymbol{q} = \left[e^{j\varphi_1}, \ldots, e^{j\varphi_M}\right]^H$, and $\bar{\boldsymbol{q}} = [\boldsymbol{q}; 1]$. Then, we get:

$$h_U = \boldsymbol{h}_{IU}^H \mathbf{\Phi} \boldsymbol{h}_{TI} + h_{TU} = \boldsymbol{q}^H \mathrm{diag}\left(\boldsymbol{h}_{IU}^H\right)\boldsymbol{h}_{TI} + h_{TU} = \boldsymbol{q}^H \boldsymbol{a}_1 + a_2, \tag{5.18}$$

$$h_E = \boldsymbol{h}_{IE}^H \mathbf{\Phi} \boldsymbol{h}_{TI} + h_{TE} = \boldsymbol{q}^H \mathrm{diag}\left(\boldsymbol{h}_{IE}^H\right)\boldsymbol{h}_{TI} + h_{TE} = \boldsymbol{q}^H \boldsymbol{a}_3 + a_4. \tag{5.19}$$

The numerator and denominator of (5.17a) are converted to:

$$\frac{1}{B}|h_U|^2 + 1 = \bar{\boldsymbol{q}}^H \boldsymbol{A}_1 \bar{\boldsymbol{q}} + \tilde{h}_U + 1, \tag{5.20}$$

$$\frac{1}{C}|h_E|^2 + 1 = \bar{\boldsymbol{q}}^H \boldsymbol{A}_2 \bar{\boldsymbol{q}} + \tilde{h}_E + 1, \tag{5.21}$$

where $\boldsymbol{A}_1 = \frac{1}{B}\begin{bmatrix} \boldsymbol{a}_1\boldsymbol{a}_1^H & \boldsymbol{a}_1 a_2^H \\ a_2\boldsymbol{a}_1^H & 0 \end{bmatrix}$, $\boldsymbol{A}_2 = \frac{1}{C}\begin{bmatrix} \boldsymbol{a}_3\boldsymbol{a}_3^H & \boldsymbol{a}_3 a_4^H \\ a_4\boldsymbol{a}_3^H & 0 \end{bmatrix}$, $\tilde{h}_U = \frac{|a_2|^2}{B}$, and $\tilde{h}_E = \frac{|a_4|^2}{C}$. Accordingly, we rewrite problem (5.17) into a more tractable problem, as follows:

$$\max_{\bar{\boldsymbol{q}}} \quad \frac{\bar{\boldsymbol{q}}^H \boldsymbol{A}_1 \bar{\boldsymbol{q}} + \tilde{h}_U + 1}{\bar{\boldsymbol{q}}^H \boldsymbol{A}_2 \bar{\boldsymbol{q}} + \tilde{h}_E + 1} \tag{5.22a}$$

$$\text{s.t.:} \quad (1-\theta)\left[B\left(\bar{\boldsymbol{q}}^H \boldsymbol{A}_1 \bar{\boldsymbol{q}} + \tilde{h}_U\right)P + \sigma_U^2\right] \geq e, \tag{5.22b}$$

$$|\boldsymbol{q}_m| = 1, \quad \forall m. \tag{5.22c}$$

The optimal solution to problem (5.22) is really not easy to find, since objective function (5.22a) is not only a non-concave function w.r.t. $\bar{\boldsymbol{q}}$ but is also a fractional function. In addition, constraint (5.22c) is a non-convex quadratic equality function for each m. Let

$\mathrm{Tr}\left(\boldsymbol{Q}\right)$ denote the trace of matrix $\boldsymbol{Q}$, and define $\boldsymbol{Q} = \bar{\boldsymbol{q}}\bar{\boldsymbol{q}}^H$ where $\boldsymbol{Q}$ is a PSD matrix and $\mathrm{rank}\left(\boldsymbol{Q}\right) = 1$. Then, problem (5.22) is transformed as follows:

$$\max_{\boldsymbol{Q}} \quad \frac{\mathrm{Tr}\left(\boldsymbol{A}_1\boldsymbol{Q}\right) + \tilde{h}_U + 1}{\mathrm{Tr}\left(\boldsymbol{A}_2\boldsymbol{Q}\right) + \tilde{h}_E + 1} \tag{5.23a}$$

$$\text{s.t.:} \quad (1-\theta)\left[B\left(\mathrm{Tr}\left(\boldsymbol{A}_1\boldsymbol{Q}\right) + \tilde{h}_U\right)P + \sigma_U^2\right] \geq e, \tag{5.23b}$$

$$\boldsymbol{Q}_{m,m} = 1, \quad \forall m \in \{1, 2, \ldots, M+1\}, \tag{5.23c}$$

$$\boldsymbol{Q} \succeq 0, \tag{5.23d}$$

$$\mathrm{rank}\left(\boldsymbol{Q}\right) = 1. \tag{5.23e}$$

By adding the two variables u, v to transform the fraction function in a way similar to the transformation from problem (5.11) to problem (5.13) in Section 5.3.4.1, problem (5.23) changes to:

$$\min_{\boldsymbol{Q},u,v} \quad -\frac{u^2}{v} \tag{5.24a}$$

$$\text{s.t.:} \quad 0 \geq u^2 - \mathrm{Tr}\left(\boldsymbol{A}_1\boldsymbol{Q}\right) - \tilde{h}_U - 1, \tag{5.24b}$$

$$0 \geq \mathrm{Tr}\left(\boldsymbol{A}_2\boldsymbol{Q}\right) + \tilde{h}_E + 1 - v, \tag{5.24c}$$

$$0 \geq \frac{e}{1-\theta} - \mathrm{Tr}\left(\boldsymbol{A}_1\boldsymbol{Q}\right)BP - \tilde{h}_U BP - \sigma_U^2, \tag{5.24d}$$

$$u, v \geq 0, \tag{5.24e}$$

$$(5.23\text{c}), (5.23\text{d}), (5.23\text{e}). \tag{5.24f}$$

Since we will also apply the FPP–SCA method to solve this problem, it is essential to find the feasible point for the final convex problem. Therefore, we can find feasible point $\boldsymbol{Q}^{(0)}$ from problem (5.24). Because variables u and v are only in constraints (5.24b) and (5.24c), we do not use constraints (5.24b) and (5.24c) when finding feasible point $\boldsymbol{Q}^{(0)}$ of problem (5.24), as follows:

$$\min_{\boldsymbol{Q}} \quad 0 \tag{5.25a}$$

$$\text{s.t.:} \quad 0 \geq \frac{e}{1-\theta} - \mathrm{Tr}\left(\boldsymbol{A}_1\boldsymbol{Q}\right)BP - \tilde{h}_U BP - \sigma_U^2, \tag{5.25b}$$

$$\boldsymbol{Q}_{m,m} = 1, \quad \forall m \in \{1, 2, \ldots, M+1\}, \tag{5.25c}$$

$$\boldsymbol{Q} \succeq 0. \tag{5.25d}$$

Solving problem (5.25), we can find feasible point $\boldsymbol{Q}^{(0)}$. Next, problem (5.24) has rank-1 constraint (5.23e). So, problem (5.24) is still non-convex. Therefore, we use the penalty method to solve the rank-1 problem, as mentioned in [152, 153]. We know that all eigenvalues of $\boldsymbol{Q}$ are non-negative, since $\boldsymbol{Q}$ is a PSD matrix. And thus, $\mathrm{Tr}\left(\boldsymbol{Q}\right) \geq \lambda_{max}\left(\boldsymbol{Q}\right)$ holds where $\lambda_{max}\left(\boldsymbol{Q}\right)$ is the maximum eigenvalue of $\boldsymbol{Q}$. Moreover, $\mathrm{Tr}\left(\boldsymbol{Q}\right) = \lambda_{max}\left(\boldsymbol{Q}\right)$ if and only if $\mathrm{rank}\left(\boldsymbol{Q}\right) = 1$. From this insight, $\left(\mathrm{Tr}\left(\boldsymbol{Q}\right) - \lambda_{max}\left(\boldsymbol{Q}\right)\right)$ should be smaller in each of the subsequent iterations. By using the penalty method, we can add the term $\eta\left(\mathrm{Tr}\left(\boldsymbol{Q}\right) - \lambda_{max}\left(\boldsymbol{Q}\right)\right)$ to objective function (5.24a), where η is the penalty factor. Problem (5.24) can be rewritten as follows:

$$\begin{aligned}
\min_{\boldsymbol{Q},u,v} \quad & -\frac{u^2}{v} + \eta\left(\mathrm{Tr}\left(\boldsymbol{Q}\right) - \lambda_{max}\left(\boldsymbol{Q}\right)\right) && \text{(5.26a)}\\
\text{s.t.:} \quad & 0 \geq u^2 - \mathrm{Tr}\left(\boldsymbol{A}_1\boldsymbol{Q}\right) - \tilde{h}_U - 1, && \text{(5.26b)}\\
& 0 \geq \mathrm{Tr}\left(\boldsymbol{A}_2\boldsymbol{Q}\right) + \tilde{h}_E + 1 - v, && \text{(5.26c)}\\
& 0 \geq \frac{e}{1-\theta} - \mathrm{Tr}\left(\boldsymbol{A}_1\boldsymbol{Q}\right)BP - \tilde{h}_U BP - \sigma_U^2, && \text{(5.26d)}\\
& u, v \geq 0, && \text{(5.26e)}\\
& \text{(5.23c)}, \text{(5.23d)}. && \text{(5.26f)}
\end{aligned}$$

In problem (5.26), the rank-1 solution of $\boldsymbol{Q}$ can be obtained when the penalty factor is large enough. However, problem (5.26) is still non-convex because functions $\left(-\frac{u^2}{v}\right)$ and $\left(-\lambda_{max}\left(\boldsymbol{Q}\right)\right)$ are non-convex. Regarding non-convex function $\left(-\frac{u^2}{v}\right)$, we perform first-order Taylor approximation for the $\frac{u^2}{v}$ function, as seen in (5.14), and convert the non-convex problem into the iterative optimization problem using the FPP–SCA method as follows:

$$\begin{aligned}
\min_{\substack{\boldsymbol{Q},u,v,\\ s_1,s_2,s_3}} \quad & \begin{array}{c} -\frac{2u^{(n)}u}{v^{(n)}} + \frac{{u^{(n)}}^2 v}{{v^{(n)}}^2} + \lambda\left(s_1+s_2+s_3\right) \\ +\eta\left(\mathrm{Tr}\left(\boldsymbol{Q}\right) - \lambda_{max}\left(\boldsymbol{Q}\right)\right) \end{array} && \text{(5.27a)}\\
\text{s.t.:} \quad & 0 \geq u^2 - \mathrm{Tr}\left(\boldsymbol{A}_1\boldsymbol{Q}\right) - \tilde{h}_U - 1 - s_1, && \text{(5.27b)}\\
& 0 \geq \mathrm{Tr}\left(\boldsymbol{A}_2\boldsymbol{Q}\right) + \tilde{h}_E + 1 - v - s_2, && \text{(5.27c)}\\
& 0 \geq \frac{e}{1-\theta} - \mathrm{Tr}\left(\boldsymbol{A}_1\boldsymbol{Q}\right)BP - \tilde{h}_U BP - \sigma_U^2 - s_3, && \text{(5.27d)}\\
& \text{(5.26e)}, \text{(5.23c)}, \text{(5.23d)}. && \text{(5.27e)}
\end{aligned}$$

Regarding non-convex function $\left(-\lambda_{max}\left(\boldsymbol{Q}\right)\right)$, we observe that $\lambda_{max}\left(\cdot\right)$ is a convex function [153]. Therefore, we can approximate the $\lambda_{max}\left(\cdot\right)$ function in an iterative manner. We review again Theorem 1 regarding the maximum eigenvalue, which is mentioned in [154].

Theorem 1. It is assumed that the PSD matrices are $\boldsymbol{X}$ and $\boldsymbol{Y}$, so $\lambda_{max}\left(\boldsymbol{X}\right) -$

$\lambda_{max}\left(\boldsymbol{Y}\right) \geq \boldsymbol{y}_{max}^{H}\left(\boldsymbol{X}-\boldsymbol{Y}\right)\boldsymbol{y}_{max}$ will be achieved where $\lambda_{max}\left(\cdot\right)$ is the maximum eigenvalue function, and $\boldsymbol{y}_{max}$ is the eigenvector according to the maximum eigenvalue of $\boldsymbol{Y}$.

From Theorem 1, we can get the inequality of PSD matrices, $\boldsymbol{Q}$ and $\boldsymbol{Q}^{(n)}$:

$$\lambda_{max}\left(\boldsymbol{Q}\right) \geq \lambda_{max}\left(\boldsymbol{Q}^{(n)}\right) + \boldsymbol{y}_{max}^{(n)H}\left(\boldsymbol{Q}-\boldsymbol{Q}^{(n)}\right)\boldsymbol{y}_{max}^{(n)}, \tag{5.28}$$

where $\boldsymbol{y}_{max}^{(n)}$ is the eigenvector according to maximum eigenvalue $\lambda_{max}\left(\boldsymbol{Q}^{(n)}\right)$ of $\boldsymbol{Q}^{(n)}$. With $\boldsymbol{w}_{max}^{(n)}$ as the eigenvector corresponding to maximum eigenvalue $\lambda_{max}\left(\boldsymbol{Q}^{(n)}\right)$, we solve the convex sub-problem in the n–th iteration as follows:

$$\begin{aligned} \min_{\substack{\boldsymbol{Q},u,v,\\ s_1,s_2,s_3}} \quad & \begin{array}{l} -\frac{2u^{(n)}u}{v^{(n)}} + \frac{{u^{(n)}}^2 v}{{v^{(n)}}^2} + \lambda\left(s_1+s_2+s_3\right) \\ +\eta\left(\mathrm{Tr}\left(\boldsymbol{Q}\right) - \boldsymbol{w}_{max}^{(n)H}\boldsymbol{Q}\boldsymbol{w}_{max}^{(n)}\right) \end{array} && (5.29a) \\ \text{s.t.:} \quad & (5.27b) - (5.27e). && (5.29b) \end{aligned}$$

Then, optimal solutions u^*, v^*, and $\boldsymbol{Q}^*$ of the n–th convex sub-problem will be used to serve as the $(n+1)$–th iteration (i.e., we update u^*, v^*, and $\boldsymbol{Q}^*$ to $u^{(n+1)}, v^{(n+1)}$, and $\boldsymbol{Q}^{(n+1)}$, respectively). We obtain $\boldsymbol{Q} = \lambda_{max}\left(\boldsymbol{Q}\right)\boldsymbol{w}_{max}\boldsymbol{w}_{max}^{H}$ when $\mathrm{Tr}\left(\boldsymbol{Q}\right) \approx \lambda_{max}\left(\boldsymbol{Q}\right)$. After that, optimal solution vector $\bar{\boldsymbol{q}} = \sqrt{\lambda_{max}\left(\boldsymbol{Q}\right)}\boldsymbol{w}_{max}$. And then, the optimal phase shifts vector $\boldsymbol{q}^*$ can be calculated as $\boldsymbol{q}^* = \left[\bar{\boldsymbol{q}}\right]_{(1:M)}$. From the definitions $\boldsymbol{\Phi} = \mathrm{diag}\left\{e^{j\varphi_1}, \ldots, e^{j\varphi_M}\right\}$ and $\boldsymbol{q} = \left[e^{j\varphi_1}, \ldots, e^{j\varphi_M}\right]^H$, we get the optimal phase shifts matrix $\boldsymbol{\Phi}^*$ from $\boldsymbol{q}^*$ with $\boldsymbol{\Phi}^* = \mathrm{diag}\left\{\boldsymbol{q}^{*H}\right\}$.

Algorithm 5.2 presents the proposed iterative algorithm based on FPP–SCA and the penalty method, while Algorithm 5.3 presents the proposed overall iterative algorithm for solving main problem (5.11).

Regarding Algorithm 5.2, we need to determine initial feasible points $\left(u^{(0)}, v^{(0)}, \boldsymbol{w}_{max}^{(0)}\right)$ of convex sub-problem (5.29) where $\boldsymbol{w}_{max}^{(0)}$ is the eigenvector corresponding to maximum eigenvalue $\lambda_{max}\left(\boldsymbol{Q}^{(0)}\right)$ which is related to the initial feasible point $\boldsymbol{Q}^{(0)}$. Fortunately, the initial feasible point $\boldsymbol{Q}^{(0)}$ can be obtained by solving the problem (5.25) (step 3 in Algorithm 5.2). After that, $u^{(0)}$ and $v^{(0)}$ can be calculated such that constraints (5.27b) and (5.27c) are satisfied, respectively (step 4 in Algorithm 5.2). Note that, in step 4, to calculate $u^{(0)}$ and $v^{(0)}$, the matrices $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ need to be calculated. As analyzed in Section 5.3.4.2, the matrices $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are involved in the calculation of the values B and C, respectively, which are also computed based on P and θ. Because Algorithm 5.2 finds the phase shift matrix $\boldsymbol{\Phi}$ by fixing P and θ, the transmit power P and the PS factor θ are

Algorithm 5.2 The FPP–SCA and penalty method–based algorithm for obtaining the phase shifts matrix ($\mathbf{\Phi}$) by fixing the transmit power (P) and the PS factor (θ)

```
Input: Channel gain of related communication links ($h_{TU}, h_{TE}, \boldsymbol{h}_{TI}, \boldsymbol{h}_{IU}, \boldsymbol{h}_{IE}$), given wireless resources ($P$, $\theta$), convergence tolerances ($\tau$, $\varepsilon_2$), penalty factor $\eta$, trade-off factor $\lambda$, required minimum harvested energy ($e$), required maximum transmit power ($P_{max}$), and $n = 0$.
Output: The optimal phase shifts: $\boldsymbol{q}^*$ and $\mathbf{\Phi}^* \Rightarrow R^*_{sec}$.
Solve problem (5.25) to find feasible point $\boldsymbol{Q}^{(0)}$, and from $\boldsymbol{Q}^{(0)}$, calculate $\boldsymbol{w}^{(0)}_{max}$
Calculate $u^{(0)}$ and $v^{(0)}$ such that $u^{(0)}$ and $v^{(0)}$ satisfy constraints (5.27b) and (5.27c), respectively, $u^{(0)} = \mathrm{Tr}\left(\boldsymbol{A}_1\boldsymbol{Q}^{(0)}\right) + \tilde{h}_U + 1$, $v^{(0)} = \mathrm{Tr}\left(\boldsymbol{A}_2\boldsymbol{Q}^{(0)}\right) + \tilde{h}_E + 1$
repeat
    Use Matlab CVX solver to find the optimal solution for problem (5.29), and obtain $\left(\boldsymbol{Q}^{(n)^*}, u^{(n)^*}, v^{(n)^*}\right)$
    $n = n + 1$
    Update $\boldsymbol{Q}^{(n)} \leftarrow \boldsymbol{Q}^{(n-1)^*}$, $u^{(n)} \leftarrow u^{(n-1)^*}$, $v^{(n)} \leftarrow v^{(n-1)^*}$, $\boldsymbol{w}^{(n)}_{max} \leftarrow \boldsymbol{w}^{(n-1)}_{max}$
until $(\mathrm{Tr}(\boldsymbol{Q}) - \lambda_{max}(\boldsymbol{Q})) \leq \tau$ and $s_1 + s_2 + s_3 \leq \varepsilon_2$
Obtain $\bar{\boldsymbol{q}}^*$ via the eigenvalue decomposition of $\boldsymbol{Q}^{(n)}$
return final $\boldsymbol{q}^*$ and $\mathbf{\Phi}^*$, calculate optimal secrecy rate $R^*_{sec}$ based on the given $P, \theta$ and $\mathbf{\Phi}^*$.
```

Algorithm 5.3 The proposed AO-based algorithm for solving problem (5.11)

```
1: Input: Channel gain of related communication links ($h_{TU}, h_{TE}, \boldsymbol{h}_{TI}, \boldsymbol{h}_{IU}, \boldsymbol{h}_{IE}$), initial phase shifts matrix $\mathbf{\Phi}$, convergence tolerances ($\tau$, $\varepsilon_1$, $\varepsilon_2$), penalty factor $\eta$, trade-off factor $\lambda$, required minimum harvested energy ($e$), required maximum transmit power ($P_{max}$), and $n = 0$.
2: Output: The optimal solutions: $P^*, \theta^*, \mathbf{\Phi}^* \Rightarrow R^*_{sec}$.
3: repeat
4:     For the given $\mathbf{\Phi}$, use Algorithm 5.1 to solve problem (5.16) and obtain $P^*$ and $\theta^*$
5:     For the given $P^*$ and $\theta^*$ obtained from Step 4, use Algorithm 5.2 to solve problem (5.29) and obtain $\mathbf{\Phi}^*$
6:     $n = n + 1$
7:     Update $\mathbf{\Phi}^{(n)} = \mathbf{\Phi}^*$
8: until convergence
9: return the optimal solution $(P^*, \theta^*, \mathbf{\Phi}^*)$, calculate optimal secrecy rate $R^*_{sec}$ based on $(P^*, \theta^*, \mathbf{\Phi}^*)$.
```

in this case the optimal transmit power and the optimal PS factor, which can be obtained from Algorithm 5.1, respectively.

For convergent analysis, similar to Algorithm 5.1, in Algorithm 5.2, the n–th optimal solution $(\boldsymbol{Q}^*, u^*, v^*)$ is a feasible point to the problem (5.29) at the $(n+1)$–th iteration, and the optimal value of problem (5.29) is non-increasing over each iteration and converges to a stationary point. Subsequently, convergence of Algorithm 5.2 is guaranteed.

5.3.4.3 The computational complexity of the proposed AO-based algorithm

In this section, we consider the computational complexity of the proposed AO-based scheme. The computational complexity is mainly from steps 4 and 5 in Algorithm 5.3, which includes the computation complexity of Algorithm 5.1 and Algorithm 5.2. As observed in Algorithm 5.1, problem (5.16) is only consisted of single non-negative variables. Therefore, the computational complexity of Algorithm 5.1 can be neglected. Besides, at step 3 of Algorithm 5.2, problem (5.25) is performed once to find the initial feasible point $\boldsymbol{Q}^{(0)}$, and thus, the computational complexity of this step can also be ignored. Finally, the computational complexity of the proposed overall algorithm is mainly from step 5 to step 9 of Algorithm 5.2 when solving problem (5.29). It is noteworthy that the convex sub-problem (5.29) can be solved by using the interior-point method. Therefore, the computational complexity can be calculated based on Theorem 3.12 [150]. According to Theorem 3.12 [150], in each iteration, when the semi-definite programming problem with an $n \times n$ PSD matrix and m constraints is given, the computational complexity is given by $\mathcal{O}\left(\sqrt{n}\log\left(1/\xi\right)\left(mn^3 + m^2n^2 + m^3\right)\right)$ where $\xi > 0$ is the solution accuracy and $\mathcal{O}\left(\cdot\right)$ is the big-O notation. For problem (5.29), since the PSD matrix $\boldsymbol{Q}$ is an $(M+1)\times(M+1)$ matrix, we can set $n = M+1$. In addition, as observing problems (5.23) and (5.27), we can set $m = 4$ due to (5.27b), (5.27c), (5.27d), and (5.23c) constraints which are related to the PSD matrix $\boldsymbol{Q}$. If we denote the number of iterations of proposed algorithm for convergence as K_1, total computational complexity of the proposed algorithm is approximated as $\mathcal{O}\left(K_1 \log\left(1/\xi\right)\left(M+1\right)^{3.5}\right)$ because of the small number of constraints $(m = 4)$.

5.3.5 Learning to optimize: the deep learning–based approach

In the previous section, we proposed the AO-based scheme, which provides optimal solution but requires high complexity and long computation time. Therefore, in this section we will consider a DL-based approach to predict the transmit power, PS factor, and the phase shifts vector. Regarding the DL-based approach, a simple DNN model is used, called a feedforward neural network (FFNN). Fig. 5.2 shows the overall flow of DL-based approach with training and running stages, where the training data are based on the solution of the AO-based scheme.

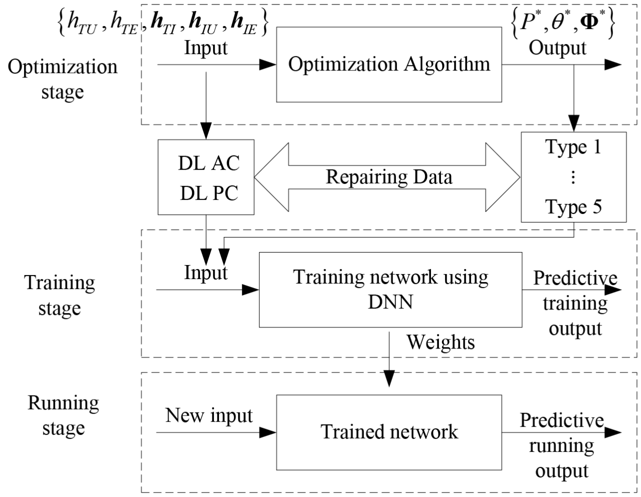

$\{h_{TU}, h_{TE}, \boldsymbol{h}_{TI}, \boldsymbol{h}_{IU}, \boldsymbol{h}_{IE}\}$
Input
Optimization stage
Optimization Algorithm
$\{P^*, \theta^*, \boldsymbol{\Phi}^*\}$
Output
DL AC
DL PC
Repairing Data
Type 1
⋮
Type 5
Training stage
Input
Training network using DNN
Predictive training output
Weights
Running stage
New input
Trained network
Predictive running output

Figure 5.2: Overall flow of the DL-based approach.

5.3.5.1 Repairing data samples and DNN training stage

In this work, the DNN-based method uses the optimal solutions obtained by the AO method as training data including transmit power, PS factor, and phase shifts. Choosing a reasonable data and DNN structures for the training process will contribute to a significant improvement in performance. Therefore, in this section, we investigate 5 types of data and DNN structures as shown in Fig. 5.3.

First, we generate N samples of the channel power gain on the related communication links $\{h_{TU}, h_{TE}, \boldsymbol{h}_{TI}, \boldsymbol{h}_{IU}, \boldsymbol{h}_{IE}\}$. By using the proposed scheme, we can get the optimal solution for transmit power P^*, PS factor θ^*, and the phase shifts vector $\boldsymbol{q}^*$ corresponding to the channel power gain of the related communication links.

In addition, we can use all channel gains of $\{h_{TU}, h_{TE}, \boldsymbol{h}_{TI}, \boldsymbol{h}_{IU}, \boldsymbol{h}_{IE}\}$ for training data of which case in the chapter is denoted as the deep learning all channels (DL AC). However, in practice it is very difficult to get the channel gain associated with the Eave' $\{h_{TE}, \boldsymbol{h}_{IE}\}$. Therefore, we consider other case where we only use channel gains

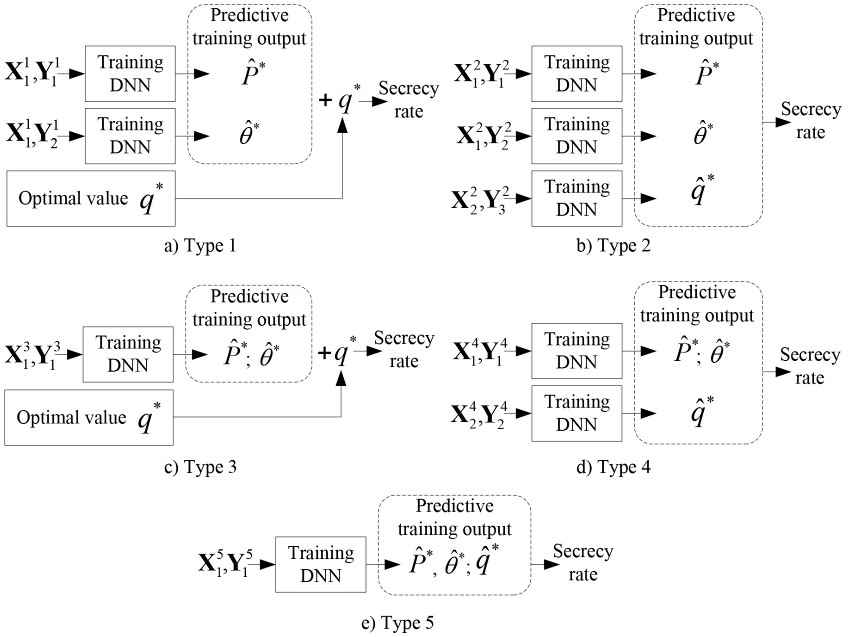

Predictive training output
$\mathbf{X}_1^1,\mathbf{Y}_1^1$
Training DNN
$\hat{P}^*$
$\mathbf{X}_1^1,\mathbf{Y}_2^1$
Training DNN
$\hat{\theta}^*$
$+q^*$
Secrecy rate
Optimal value q^*
a) Type 1
Predictive training output
$\mathbf{X}_1^2,\mathbf{Y}_1^2$
Training DNN
$\hat{P}^*$
$\mathbf{X}_1^2,\mathbf{Y}_2^2$
Training DNN
$\hat{\theta}^*$
$\mathbf{X}_2^2,\mathbf{Y}_3^2$
Training DNN
$\hat{q}^*$
Secrecy rate
b) Type 2
Predictive training output
$\mathbf{X}_1^3,\mathbf{Y}_1^3$
Training DNN
$\hat{P}^*; \hat{\theta}^*$
$+q^*$
Secrecy rate
Optimal value q^*
c) Type 3
Predictive training output
$\mathbf{X}_1^4,\mathbf{Y}_1^4$
Training DNN
$\hat{P}^*; \hat{\theta}^*$
$\mathbf{X}_2^4,\mathbf{Y}_2^4$
Training DNN
$\hat{q}^*$
Secrecy rate
d) Type 4
Predictive training output
$\mathbf{X}_1^5,\mathbf{Y}_1^5$
Training DNN
$\hat{P}^*, \hat{\theta}^*; \hat{q}^*$
Secrecy rate
e) Type 5

Figure 5.3: Five types of data and DNN structures.

of $\{h_{TU}, \boldsymbol{h}_{TI}, \boldsymbol{h}_{IU}\}$, which is denoted as the deep learning partial channels (DL PC). In addition, different training data structures are built for the training stage, which are denoted as Type 1 to Type 5. For simplicity, instead of using the phase shifts vector as a complex number (i.e., $\boldsymbol{q}^* = \left[e^{j\varphi_1^*}, \ldots, e^{j\varphi_M^*}\right]^H$), we convert it into the real part and the imaginary part of the phase shifts vector, denoted $\mathrm{Re}\left(\boldsymbol{q}^*\right)$ and $\mathrm{Im}\left(\boldsymbol{q}^*\right)$, respectively.

We denote the channel gain matrix as $\mathbf{X}$, with $\mathbf{Y}$ being the output matrix of the optimization solution, which is the optimal power allocation, the PS factor, and the real and imaginary parts of the phase shifts vector. The size and structure of the training data will depend on the type of structure. For construction of the data for the training stage, we present the case under DL AC. DL PC is done similarly, but only with the channel gain for $\{h_{TU}, \boldsymbol{h}_{TI}, \boldsymbol{h}_{IU}\}$. The training data structures are described as follows.

- **Type 1 (Fig. 5.3a):** the optimal transmit power and PS factor are constructed separately (and thus, will also be trained separately). The optimal phase shifts vector is used and combined with the predictive training output for transmit power and the PS factor in order to calculate the secrecy rate. Then, the input and output matrices are given as follows:

$$\mathbf{X}_1^1 = \left[|h_{TU}|^2 ; |h_{TE}|^2 ; |\boldsymbol{h}_{TI}|^2 ; |\boldsymbol{h}_{IU}|^2 ; |\boldsymbol{h}_{IE}|^2 \right]_{(1:N)} , \tag{5.30}$$

$$\mathbf{Y}_1^1 = [P^*]_{(1:N)} , \tag{5.31}$$

$$\mathbf{Y}_2^1 = [\theta^*]_{(1:N)} . \tag{5.32}$$

The superscripts of $\mathbf{X}$ and $\mathbf{Y}$ indicate the type of training data structure from Type 1 to Type 5 whereas the subscripts are the distinctive numbering. Because $h_{TU}, h_{TE} \in \mathbb{C}^{1\times 1}$ and $\boldsymbol{h}_{TI}, \boldsymbol{h}_{IU}, \boldsymbol{h}_{IE} \in \mathbb{C}^{M\times 1}$ where M is the number of reflecting elements. In addition, if the number of samples for training data is N, the training input and output data size of Type 1 will be $\mathbf{X}_1^1 \in \mathbb{C}^{(2+3M)\times N}$ and $\mathbf{Y}_1^1, \mathbf{Y}_2^1 \in \mathbb{C}^{1\times N}$, respectively.

- **Type 2 (Fig. 5.3b):** the optimal transmit power, the PS factor, and the phase shifts vector are constructed separately. Because the real part and imaginary part of the phase shifts vector are used, we also need to convert channel gain to the real part and imaginary part, which ensures the channel gain and the phase shifts vector use the same dimension of the samples for the training stage. Then, the input and output matrices are given as follows:

$$\mathbf{X}_1^2 = \mathbf{X}_1^1, \tag{5.33}$$

$$\mathbf{Y}_1^2 = \mathbf{Y}_1^1, \tag{5.34}$$

$$\mathbf{Y}_2^2 = \mathbf{Y}_2^1, \tag{5.35}$$

$$\begin{aligned} \mathbf{X}_2^2 = [& \{\mathrm{Re}\left(h_{TU}\right), \mathrm{Im}\left(h_{TU}\right)\} ; \{\mathrm{Re}\left(h_{TE}\right), \mathrm{Im}\left(h_{TE}\right)\} ; \{\mathrm{Re}\left(\boldsymbol{h}_{TI}\right), \mathrm{Im}\left(\boldsymbol{h}_{TI}\right)\} ; \\ & \{\mathrm{Re}\left(\boldsymbol{h}_{IU}\right), \mathrm{Im}\left(\boldsymbol{h}_{IU}\right)\} ; \{\mathrm{Re}\left(\boldsymbol{h}_{IE}\right), \mathrm{Im}\left(\boldsymbol{h}_{IE}\right)\}]_{(1:N)} , \end{aligned} \tag{5.36}$$

$$\mathbf{Y}_3^2 = [\{\mathrm{Re}\left(\boldsymbol{q}^*\right), \mathrm{Im}\left(\boldsymbol{q}^*\right)\}]_{(1:N)} . \tag{5.37}$$

Similarly, the training input and output data size of Type 2 based on (5.33), (5.34), and (5.35) are $\mathbf{X}_1^2 \in \mathbb{C}^{(2+3M)\times N}$ and $\mathbf{Y}_1^2, \mathbf{Y}_2^2 \in \mathbb{C}^{1\times N}$, respectively. Due to the real and imaginary parts of the phase shift and the size of the phase shift, $\boldsymbol{q}$, depends on the number of elements M, the training data size of $\mathbf{X}_2^2$ and $\mathbf{Y}_3^2$ are $\mathbf{X}_2^2 \in \mathbb{C}^{(2+3M)\times 2N}$ and $\mathbf{Y}_3^2 \in \mathbb{C}^{M\times 2N}$, respectively.

- **Type 3 (Fig. 5.3c):** the optimal transmit power and PS factor are constructed and used for the training data, whereas the optimal phase shifts vector is used. The input and output matrices are:

$$\mathbf{X}_1^3 = \mathbf{X}_1^1, \tag{5.38}$$

$$\mathbf{Y}_1^3 = \left[P^*; \theta^*\right]_{(1:N)}. \tag{5.39}$$

The training input and output data size of Type 3 are $\mathbf{X}_1^3 \in \mathbb{C}^{(2+3M)\times N}$ and $\mathbf{Y}_1^3 \in \mathbb{C}^{2\times N}$, respectively.

- **Type 4 (Fig. 5.3d):** the optimal transmit power and PS factor are constructed and used for the training data, whereas the optimal phase shifts vector is constructed separately for the training stage. The input and output matrices are:

$$\mathbf{X}_1^4 = \mathbf{X}_1^1, \tag{5.40}$$

$$\mathbf{Y}_1^4 = \mathbf{Y}_1^3, \tag{5.41}$$

$$\mathbf{X}_2^4 = \mathbf{X}_2^2, \tag{5.42}$$

$$\mathbf{Y}_2^4 = \mathbf{Y}_3^2. \tag{5.43}$$

The training input and output data size of Type 4 are $\mathbf{X}_1^4 \in \mathbb{C}^{(2+3M)\times N}$, $\mathbf{X}_2^4 \in \mathbb{C}^{(2+3M)\times 2N}$, $\mathbf{Y}_1^4 \in \mathbb{C}^{2\times N}$, and $\mathbf{Y}_2^4 \in \mathbb{C}^{M\times 2N}$.

- **Type 5 (Fig. 5.3e):** the optimal transmit power, PS factor, and the phase shifts vector are constructed and used for the training data for the training stage. The input and output matrices are:

$$\mathbf{X}_1^5 = \mathbf{X}_2^2, \tag{5.44}$$

$$\mathbf{Y}_1^5 = \left[\left\{P^*, \theta^*\right\}; \left\{\mathrm{Re}\left(\boldsymbol{q}^*\right), \mathrm{Im}\left(\boldsymbol{q}^*\right)\right\}\right]_{(1:N)}. \tag{5.45}$$

The training input data size of Type 5 is $\mathbf{X}_1^5 \in \mathbb{C}^{(2+3M)\times 2N}$. According to (5.45), the training data output size is $\mathbf{Y}_1^5 \in \mathbb{C}^{(1+M)\times 2N}$.

Next, the training data are trained by the DNN using backpropagation. The scaled conjugate gradient algorithm is used in the training process to optimize the MSE. To perform backpropagation in the training stage, two activation functions are used: $purelin\left(\cdot\right)$ is used for the output layer, whereas $tansig\left(\cdot\right)$ is used for hidden layers; they are calculated as follows:

$$tansig\left(x\right) = \frac{2}{1+e^{-2x}} - 1, \tag{5.46}$$

$$purelin\left(x\right) = x. \tag{5.47}$$

5.3.5.2 DNN Running stage

In the running stage, we also generate channel matrices $\mathbf{Z}$ for the input layer according to the type of training data structure but with K samples for channel power gain. Channel gain is generated the same way as in the training stage. Then, for the run data, the well-trained network is loaded for channel matrix $\mathbf{Z}$. Finally, the output layer produces the running predictive optimal value, which includes the predictive optimal transmit power $\hat{P}$, PS factor $\hat{\theta}$, real part $\mathrm{Re}\left(\hat{\boldsymbol{q}}\right)$ and imaginary part $\mathrm{Im}\left(\hat{\boldsymbol{q}}\right)$ of predictive optimal phase shifts vector $\hat{\boldsymbol{q}}$ according to the type of training data structure.

Fig. 5.3 shows the 5 types of data and DNN structures where Type 1 and Type 3 structures only estimate power $\hat{P}^*$ and PS factor $\hat{\theta}^*$ by using DNN while the optimal value of the phase shift $\boldsymbol{q}^*$ directly is calculated. Specifically, in the running stage of Type 1 and Type 3 structures, the value of the phase shift is not available so, for the new channel gain input, we must use to Algorithm 5.2 to get optimal $\boldsymbol{q}^*$. That is, we can get the phase shifts matrix by using Algorithm 5.2 while fixing the transmit power and the PS factor as the estimated power $\hat{P}^*$ and PS factor $\hat{\theta}^*$. Even we can set the estimated value of $\hat{\boldsymbol{q}}^*$ by DNN as initial value for Algorithm 5.2. On the other hand, Type 2, Type 4, and Type 5 structures estimate transmit power $\hat{P}^*$, PS factor $\hat{\theta}^*$, and phase shift $\hat{\boldsymbol{q}}^*$ using DNN structures.

5.4 Simulation results and discussion

First, we set the necessary parameters for the optimization algorithm and the DNN. Then, the numerical results for the average secrecy rate (ASR) from changing the transmitter power and the number of IRS reflecting surfaces are provided. We also consider the effect on the ASR of circuit noise at the UE as well as factors affecting channel gain (such as the vertical distance between the UE and the IRS, as well as the path loss exponents). Regarding the DL-based approach, we use the DL scheme for the ASR based on changes to the required minimum harvested energy with different structures of the training data. With regard to the proposed optimization-based approach, the solution to the problem can be obtained, and it converges to the optimal value through a number of iterations. Meanwhile, the proposed DL-based approach shows the ability to approximate the response that is produced by the optimization algorithm. In our work, benchmark schemes are used, including a scheme without an IRS, a random phase shifts scheme, and the equal PS–factor scheme. The scheme without an IRS only finds the optimal resource allocation (i.e., only the optimal transmit power and PS factor). The random phase shifts scheme reuses the optimal resource allocation from the scheme without an IRS, and combines it with the random phase shifts vector to calculate the system secrecy rate. The equal PS–factor scheme uses the optimization algorithm without IRS to solve the problem, and the PS factor is fixed so that the gain of the PS factor across the ID and EH streams is equal, i.e., the PS factor is set to 0.5 ($\theta = 0.5$).

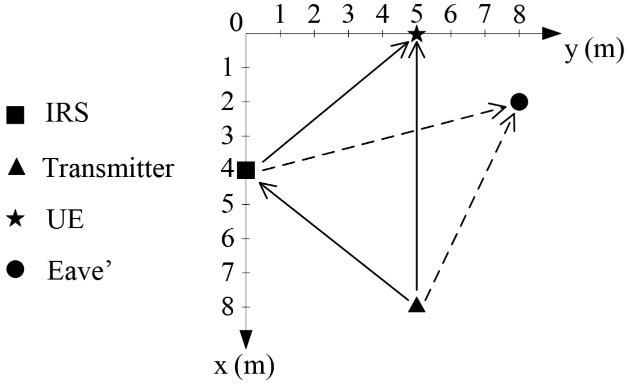


Figure 5.4: The horizontal coordinates of the IRS, the transmitter, the UE, and the Eave' on the $x - y$ plane.

5.4.1 The neural network configuration and simulation parameters

In our work, we setup the system at small scale on a three-dimensional Cartesian coordinate system. The reference (center) point of the IRS is located at $\mathbf{w}_I = [4,0]^T$ m and $z_I = 5$ m. The transmitter, the UE, and the Eave' are located in the $x-y$ plane, and thus, $z_T = z_U = z_E = 0$ m. The horizontal coordinates of the transmitter, the UE, and the Eave' on the $x-y$ plane are denoted by $\mathbf{w}_T = [8,5]^T$ m, $\mathbf{w}_U = [0,5]^T$ m, and $\mathbf{w}_E = [2,8]^T$ m, respectively. For visualization, in addition to the altitude of the IRS ($z_I = 5$ m), the locations of the IRS, the transmitter, the UE, and the Eave' on the $x-y$ plane are shown in Fig. 5.4.

Table 5.3: Hyper-parameters descriptions

Symbol	Description	Value
ρ_l	path loss at reference distance $D_0 = 1$ m	-20 dB
Δ	antenna separation	$\frac{\lambda_c}{2}$
$\alpha_{TU}, \alpha_{TE}, \alpha_{TI},$	path loss exponents of the communication links	2.5, 3, 2,
α_{IU}, α_{IE}		2, 2.5
$\beta_{TI}, \beta_{IU}, \beta_{IE}$	Rician factors of the communication links	3 dB
σ_U^2	noise variance from antenna at the UE	-70 dBW
δ_U^2	noise variance of circuit at the UE	-50 dBW
σ_E^2	noise variance from antenna at the Eave'	-60 dBW
λ	trade-off factor[a]	100
ε_1	convergence tolerance	10^{-4}
ε_2	convergence tolerance	10^{-3}
τ	convergence tolerance	10^{-4}
η	penalty factor	10^5
N	the number of training samples	1000
K	the number of testing samples	100

[a] The trade-off factor, λ, is usually of great value (usually $\lambda \gg 1$), which can help ensure that if the optimization problem has a feasible solution, then the problem will still remain in the feasible region on subsequent iterations [135].

Regarding the channel model, the CSCG random variable distribution was used for the channel gain of the T–U link ($\tilde{h}_{TU}$), the T–E link ($\tilde{h}_{TE}$), and the NLoS components of related communication links ($\boldsymbol{h}_{TI}^{NLoS}, \boldsymbol{h}_{IU}^{NLoS}, \boldsymbol{h}_{IE}^{NLoS}$). Depending on the requirements of the simulation, the required maximum transmit power (P_{max}) and the required minimum harvested energy (e) were specifically provided in each simulation. For the DL-based approach, an FFNN model with four layers was used, which has not only input and output layers but also two hidden layers. We set 20 neurons for each hidden layer. The other simulation parameters are shown in Table 5.3.

5.4.2 The secrecy rate performance under various configurations

In this section, system performance is compared under different settings. First, we investigate the convergence property. Then, we check the effect of the number of IRS reflecting elements in terms of the ASR. After that, the ASR based on changing the required maximum transmitter power is investigated. Finally, we validate the ASR under factors affecting channel gain (such as the vertical distance between the UE and the IRS, as well as the path loss exponent of the T–U link).

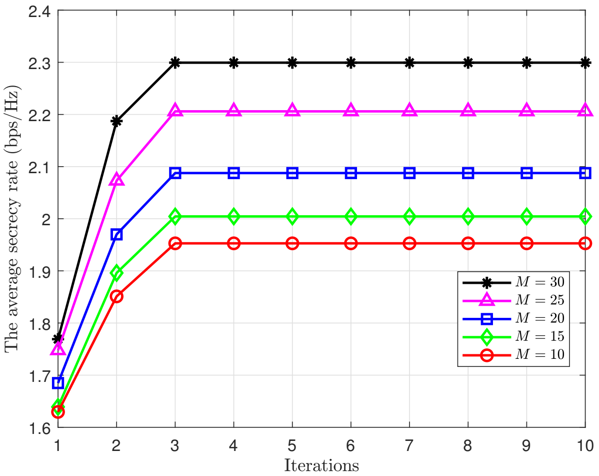


Figure 5.5: The ASR of the proposed AO algorithm based on the number of iterations when changing the number of reflecting elements, M.

Fig. 5.5 shows the convergence property on the ASR according to the number of iterations under our proposed scheme with an IRS. We observed that the ASR increased rapidly and reached an optimal solution between the first and the third iteration. Fig. 5.5

also shows the improvement of the ASR when the IRS's reflecting surfaces increases in number. That is the result of using and optimizing the IRS's phase shifts, contributing to the enhancement of the received signal at the UE and the weakening of the received signal at Eave' when the number of IRS reflecting surfaces increases. To see this clearly, in Fig. 5.6 we checked the ASR from different schemes based on the number of IRS elements, M, when the requirements of the UE's harvested energy and the transmitter's power are fixed at $e = -54$ dBW and $P_{max} = 100$ W, respectively.

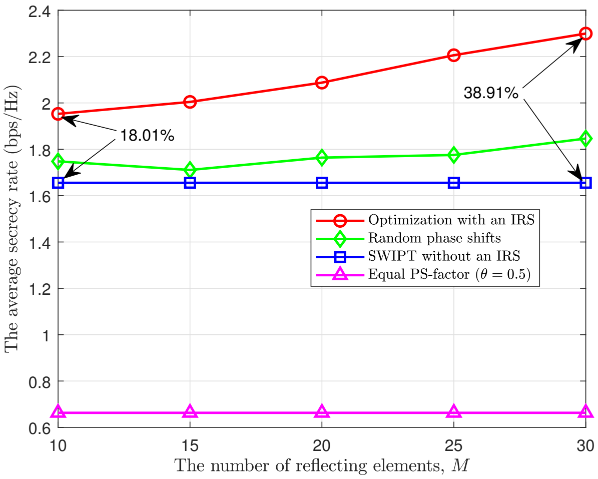


Figure 5.6: The ASR of the different schemes according to the number of reflecting elements, M, when the required minimum harvested energy is $e = -54$ dBW and the required maximum transmit power is $P_{max} = 100$ W.

Fig. 5.6 shows that the ASR increased significantly when the number of IRS reflecting elements increased. Specifically, the proposed IRS scheme improved the ASR from 18.01% to 38.91% when increasing the IRS reflecting elements from 10 to 30. This is because as the IRS reflecting elements increase in number, the signals from the IRS become dominant at the UE and degrade for the Eave'. Fig. 5.6 also shows that the proposed scheme outperforms the random phase shifts scheme and the scheme without an IRS, which results from using and optimizing the IRS phase shifts. By optimizing the phase shifts, the signals reflected by the IRS can be optimized and combined with the signals directly from the T–U and T–E links to enhance or degrade the signals obtained at the UE and the Eave', respectively, thus contributing to strengthening the system secrecy rate to a higher degree, compared to not using the IRS. The performance of the random phase shifts scheme is less efficient than the proposed scheme, but it is better than the optimization scheme without an IRS, because the random phase shifts scheme reuses the optimal transmit power and PS factor from the optimization scheme without the IRS, along with random phase shifts, to calculate the secrecy rate. A note on random phase shift: although the ASR also tends to increase when the number of reflecting elements increases, in random phase shifts that are not properly optimized, the performance can not only be worse than the optimization scheme with an IRS but can be even worse when increasing the number of reflecting elements, for example, when $M = 10$ and $M = 15$, as shown in Fig. 5.6. The equal PS–factor scheme provides the lowest ASR because it can only achieve the optimal secrecy rate based on the optimal transmit power while the PS factor is fixed at $\theta = 0.5$. Note that when the number of reflecting elements increases, the ASR from optimization scheme without an IRS remains unchanged, since the IRS is not used.

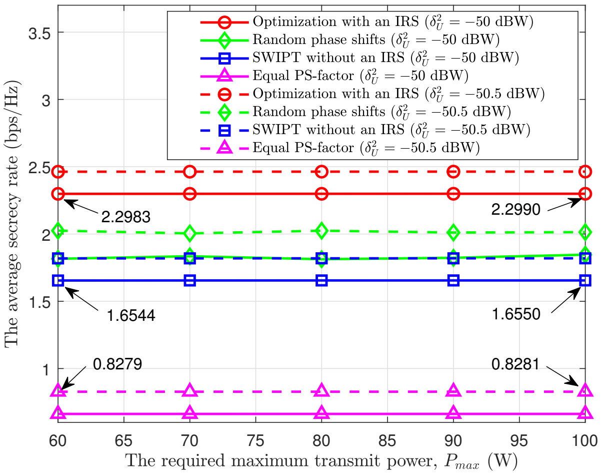


Figure 5.7: The ASR of the different schemes according to the required maximum transmit power when the number of reflecting elements is $M = 30$ and the required minimum harvested energy is $e = -54$ dBW.

In Fig. 5.7, we consider the effect of the required maximum transmit power at the transmitter on the ASR of the schemes. In this case, the required maximum transmit power is based on the values $P_{max} \in \{60, 70, 80, 90, 100\}$ W, while the required minimum harvested energy is fixed at $e = -54$ dBW, and the number of reflecting elements is 30. As observed in Fig. 5.7, again, the optimization scheme with the IRS achieves the highest ASR, while the equal PS–factor scheme achieves the lowest ASR. In addition, although the required maximum transmit power increases, in our scenario, due to the impact of noise and channel gain, the achievable rate at the UE changes relatively little. Therefore, Fig. 5.7 shows that the ASR increases very little. For the slight increase in terms of ASR according to the required maximum transmit power, it is not necessary to use too much power. Therefore, in operation, we can choose the appropriate transmit power to ensure performance and not consume too many resources. Fig. 5.7 also shows that the ASR would be improved by reducing circuit noise at the UE. Reducing processing noise at the UE is completely achievable as science and technology develop more and more.

Let d_v denote the vertical distance between the UE and the IRS, and consider UEs at the following locations: $\mathbf{w}_1^U = [0,1]^T, \mathbf{w}_2^U = [0,3]^T, \mathbf{w}_3^U = [0,5]^T$, and $\mathbf{w}_4^U = [0,7]^T$ m, as shown in Fig. 5.8. This also means that the vertical distance between the UE and the IRS is considered based on $d_v \in \{1,3,5,7\}$ m.

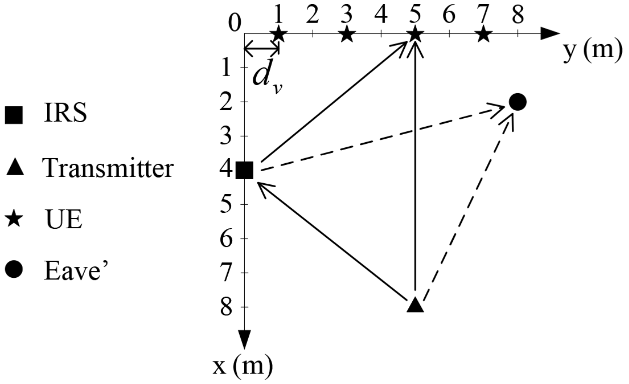


Figure 5.8: The vertical distance between the UE and the IRS, d_v, and the locations of the UEs (star symbols) on the $x-y$ plane.

Fig. 5.9 shows the ASR of different schemes when changing the vertical distance d_v between the UE and the IRS. As observed in Fig. 5.9, with regard to schemes that do not use an IRS (i.e., the optimization scheme without an IRS and the equal PS–factor scheme), the best ASR is achieved when the UE is closest to the transmitter (i.e., when $d_v = 5$ m). Conversely, the ASR decreases if the UE is farther away from the transmitter (i.e., when $d_v = 1, 3$, and 7 m). This is understandable since the channel is modeled according to the Rayleigh model, and as a result, the greater the distance between the transmitter and the UE, the more the channel is attenuated. Therefore, the signal received at the UE is reduced, and the secrecy rate decreases. One thing to note is that when the distance between the transmitter and the UE is equal (when $d_v = 3$ m and $d_v = 7$ m), channel gain, h_{TU} at $d_v = 3$ m and $d_v = 7$ m is the same owing to the Rayleigh fading channel model, as seen in (1), so the ASR gives the same result at $d_v = 3$ m and $d_v = 7$ m. Regarding the optimization scheme with an IRS, the best ASR is achieved at $d_v = 3$ m. This shows that the closer the UE is to the IRS, the more the reflected signal from the IRS is enhanced, resulting in a stronger signal at the UE. However, as mentioned above, the signal strength at the UE also depends on it being a direct signal from the transmitter, which shows that when the UE is farther from the transmitter, the direct signal from the T–U link decreases. Therefore, when the UE is close to the IRS at a certain distance (for example, at $d_v = 1$ m from the result in Fig. 5.9), the combination of the T–U link's direct signal and the I–U link's reflected signal is no longer optimal, resulting in the ASR decreasing at $d_v = 1$ m.

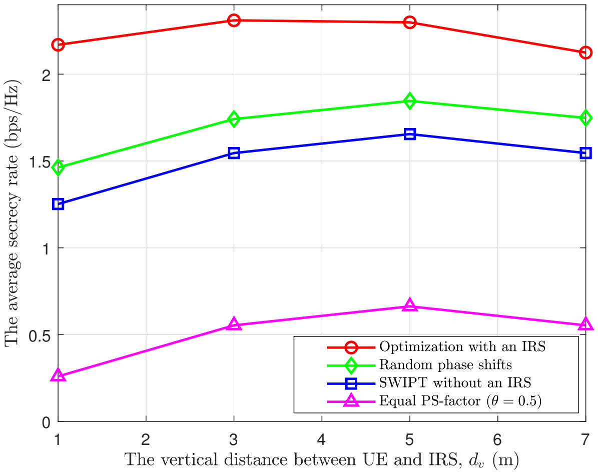


Figure 5.9: The ASR of the different schemes according to the vertical distance between the UE and the IRS d_v when required minimum harvested energy is $e = -54$ dBW, required maximum transmit power is $P_{max} = 100$ W, and the number of reflecting elements is $M = 30$.

Fig. 5.10 shows the ASR of the different schemes according to the path loss exponent of the T–U link, α_{TU}. Usually, the path loss exponent has a range between 1.5 and 5 [155], so we considered path loss exponent values from 1.5 to 3. In general, the ASR tends to decrease as the path loss exponent increases, and is even less than 0 when the path loss exponent is high ($\alpha_{TU} = 3$) for a low-performance scheme like the equal PS–factor scheme. This is caused by a decrease in the T–U channel gain as the path loss exponent increases accordingly. As a result, the signal received at the UE also decreases, leading to a decrease in the secrecy rate. Fig. 5.10 also shows that with the help of the IRS, the ASR of the optimization scheme with an IRS decreases more slowly than the other schemes. Again, the proposed scheme with an IRS outperforms the other schemes. It is noteworthy that, when the pathloss exponent is small (e.g., the path loss exponent is 1.5), the difference in the channel gain value, h_U, between the scheme without an IRS and scheme with an IRS is insignificant under other conditions unchanged. Therefore, the secrecy rate between these two schemes may be approximately, or even the secrecy rate of the proposed scheme may be smaller than that of the scheme without an IRS. In addition, the secrecy rate of the random phase shift scheme in our work is calculated based on the optimal transmit power and PS factor of the scheme without an IRS and the random phase shifts vector. Therefore, the secrecy rate of the random phase shift scheme may be greater than that of the scheme with an IRS when the path loss exponent is small. Fortunately, the path loss exponent is generally greater than 2 for obstructions to the propagation of the energy of an electromagnetic wave [155]. Thus, Fig. 5.10 shows that the proposed scheme provides an acceptable performance when an appropriate path loss exponent value is used (e.g., path loss exponent values such as 2 and 2.5 in common transmission environments).

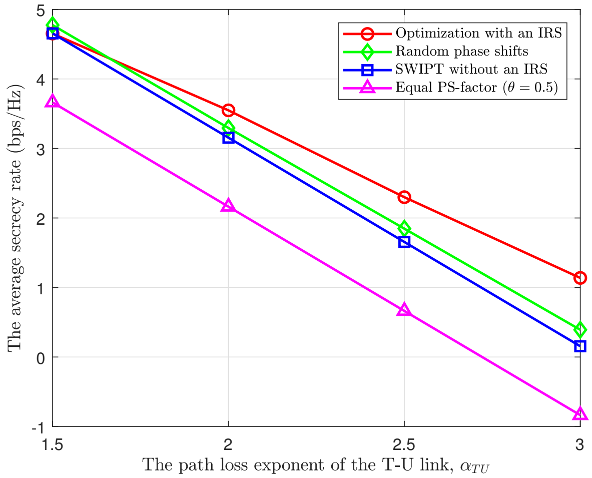


Figure 5.10: The ASR of the different schemes according to the path loss exponent of the T–U link, α_{TU}, when required minimum harvested energy is $e = -54$ dBW, required maximum transmit power is $P_{max} = 100$ W, and the number of reflecting elements is $M = 30$.

5.4.3 The DL-based approach to computation time performance

We further inspect the approximation of the DL scheme on the ASR based on changes in the required minimum harvested energy. In addition, system performance in terms of computation time under the different schemes is evaluated in this section.

After finding a solution via CVX is complete, the CVX tool can summarize the result into the *cvx_status* string variable. The CVX solver has several status levels, like *solved, unbounded, infeasible*, or even *failed*, and many others [107]. Therefore, although the CVX solver can effectively solve the convex optimization problem, the problem may still reach an infeasible solution where the CVX solver cannot find the optimal solution to the optimization problem (i.e., the *cvx_status* is **not** *solved*). As a result, although a large amount of channel gain in related communication links is generated, the optimal solution may not be found for a certain channel gain. Furthermore, in this work, the CVX tool was executed in each of the iterations in the FPP–SCA iterative approach where the solution converges to an optimal value after a number of iterations. Hence, it is very time-consuming to generate huge amounts of samples for training data. Therefore, in this work, to benefit from the efficiency of the DL approach, we try to generate about 1000 samples from feasible solutions of the optimization algorithm for training data, with about 100 samples for running data.

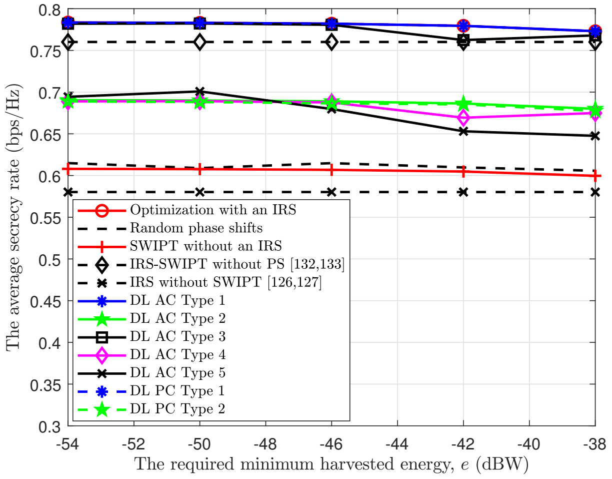


Figure 5.11: The ASR of DL schemes according to the required minimum harvested energy when the required maximum transmit power is $P_{max} = 100$ W, and the number of reflecting elements is $M = 10$.

Fig. 5.11 shows the average secrecy rate (ASR) of the DL-based approach according to the required minimum harvested energy when required maximum transmit power is $P_{max} = 100$ W and the number of reflecting elements is $M = 10$. Here, we observe ASR for the different training data structures as well as for cases where all the channels (DL AC) and partial channels (DL PC) are utilized respectively. From Fig. 5.11, we observed the following things. First, the ASR decreases slightly as the required minimum harvested energy increases. This is because as the required minimum harvested energy is increased, the PS factor should be reduced to ensure more harvesting energy, as shown in the constraint (5.11b). Subsequently, a decrease of PS factor causes the UE's achievable rate to be decreased, which results in a decrease in the ASR.

Second, DL AC Type 1 and DL AC Type 3 provide near optimal value of ASR, compared to the AO method. However, DL AC Type 1 gives better performance than DL AC Type 3 since DL AC Type 3 uses one DNN for estimating transmit power $\hat{P}^*$ and PS factor $\hat{\theta}^*$ while DL AC Type 1 uses two DNNs. It is noteworthy that optimal value of the phase shift should be calculated in the case of DL AC Type 1 and DL AC Type 3.

Third, DL AC Type 2, DL AC Type 4, and DL AC Type 5 where transmit power $\hat{P}^*$ and PS factor $\hat{\theta}^*$, and phase shift $\hat{\boldsymbol{q}}^*$ are estimated by DNN, provide less performance than DL AC Type 1 and DL AC Type 3. Among DL AC Type 2, DL AC Type 4, and DL AC Type 5, the DL AC Type 2 provides the best performance since it utilizes three DNNs for estimating transmit power $\hat{P}^*$ and PS factor $\hat{\theta}^*$, and phase shift $\hat{\boldsymbol{q}}^*$, respectively. However, the performances of all DL AC Type 2, DL AC Type 4, and DL AC Type 5 are better than those of the optimization scheme without an IRS and random phase shifts.

Fourth, in practice, it is very difficult to obtain the channel gain associated with the Eave' $\{h_{TE}, \boldsymbol{h}_{IE}\}$. In the work, DL PC was considered as DL PC Type 1 and DL PC Type 2. From Fig. 5.11, interestingly, it is observed that DL PC achieves the similar performance to DL AC. Subsequently, the proposed DL PC Type 1 and DL PC Type 2 have practical applications since the channel gains from Eave's are not required in advance for obtaining transmit power $\hat{P}^*$ and PS factor $\hat{\theta}^*$, and phase shift $\hat{\boldsymbol{q}}^*$.

Finally, Fig. 5.11 also shows the secrecy rate of the proposed scheme compared to that of the existing IRS-aided secure transmission schemes. As observed, the proposed scheme outperforms the IRS-SWIPT without PS scheme [132, 133] and IRS without SWIPT scheme [126, 127]. This is due to the influence of SWIPT as well as the PS factor. The IRS-SWIPT without PS scheme [132, 133] does not use the PS factor. Therefore, the secrecy rate of this scheme tends to increase slightly as the required minimum harvested energy increases, but it is almost negligible. It should be noted that IRS without the SWIPT scheme [126, 127] does not use the SWIPT system, and therefore it is not affected by the required minimum harvested energy. As a result, the secrecy rate of this scheme remains unchanged.

Fig. 5.12 shows the computation time in running stage of Type 1, Type 2, and Type 3 DL AC schemes compared with optimization and benchmark schemes. The optimization scheme with an IRS takes a long time to implement even though the number of samples of channel gain in the related communication links is small (only 100). This is because the AO algorithm uses an alternative method to find solutions. The equal PS–factor scheme skips some calculations related to the PS factor because the PS factor is fixed at $\theta = 0.5$. Therefore, the computation time of this scheme is less than the optimization scheme without an IRS. Along with performance close to that of the proposed AO algorithm as shown in Fig. 5.11, Type 1 and Type 3 clearly improve computation time compared to the proposed AO algorithm (i.e., scheme with an IRS) in Fig. 5.12. This is because when DL is applied, the optimal phase shifts can be achieved only by using Algorithm 5.2. Type 2, Type 4, and Type 5 provide low running times. However, as observed in Fig. 5.11, the performance of these types is better than that of the scheme without IRS and worse than that of the optimal scheme with an IRS.

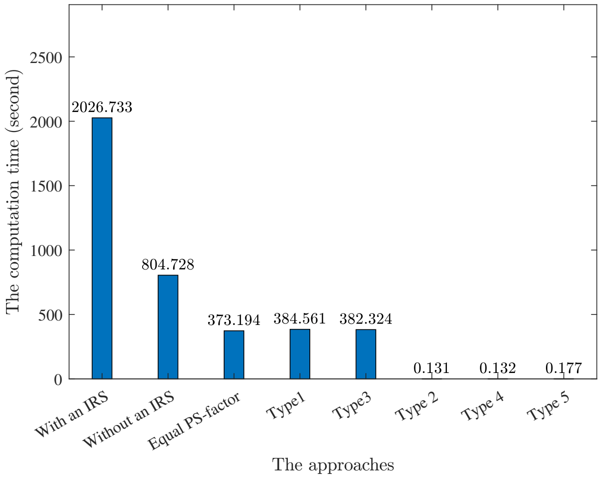


Figure 5.12: The computation time comparison when the required maximum transmit power is $P_{max} = 100$ W and the number of reflecting elements is $M = 30$.

5.5 Conclusion

In our work, an IRS-assisted secure transmission maximization scheme for SWIPT systems with a PS scheme is considered. We first aim to maximize the system secrecy rate by finding the optimal transmitter power, UE PS factor, and IRS phase shifts while satisfying the requirements of energy harvesting at the user and transmit power at the transmitter. For solving the optimization problem, we invoked an AO algorithm in which an FPP-SCA iterative algorithm and a penalty method are used to find the optimal solutions in an alternating manner. The simulation results show that the scheme helped by the IRS achieves a significant improvement in terms of ASR, compared to the scheme without an IRS. Then, we proposed a DL-based approach to improve computation performance. The comparison results showed that the DL-based approach not only provided performance similar to that of the optimization algorithm but significantly improved computation time. For future work, our work can be extended to a multiple-antenna transmitter and even to multiple PS users. In addition, with the benefits of the unmanned aerial vehicles (UAV) in significantly improving capacity, throughput and reliability, the combination of UAVs with IRS opens promising research directions. However, this also brings many challenges such as channel modeling, channel estimation, and especially new dimensions, like the UAV's location and trajectory. In addition, to meet the real-time processing requirements of large-scale heterogeneous communication systems, deep Q networks and deep deterministic policy-gradient algorithms in deep reinforcement learning are potential solutions to solve our problem. Even so, further studies on these combinations are worth pursuing as one of our future works.

Chapter 6

Summary of Contributions and Future Works

6.1 Introduction

The research motivations, the problems of CRN and SWIPT systems, and AI-based solution have been provided in previous chapters. Then, 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 AI on radio resource management for advanced WCNs, CRN and SWIPT, in which optimization algorithm, RL and DL are applied. These techniques are not only enhance the performance of the network but significantly improve the computation performance. The main contributions of this research are summarized as follows:

Firstly, we build a software-defined radio testbed-based spectrum sensing system to evaluate sensing performance in a real environment. In particular, a new block of energy detection is generated by using an out-of-tree module from GNU Radio. We also integrate CR into the cloud computing paradigm by connecting to cloud through ThingSpeak such that we can store, process, and share the sensing information to the cloud server. In addition, we utilize a video transmission with spectrum-sensing system. In this case, spectrum sensing is implemented on both transmitter and receiver. Specifically, at the transmitter, if the sensed channel is free, a video signal is transmitted on the sensed frequency. Conversely, if the channel is busy, a video signal will be transmitted on the next frequency. At the receiver, if the sensed channel is detected, the signal is demodulated to reproduce the transmitted video signal. In this research, the video can be transmitted both H.264 and MP4 formats. The testbed implementations show that ED-based spectrum sensing and video transmission is done efficiently with appropriate parameters. In addition, by connecting to Thing Speak, the sensed information at a local CR node can be stored, processed, and shared more efficiently in a centralized way.

Secondly, we consider the jamming attack scenario in multi-channel CRN where an SU is transmitting data to a receiver SU while multiple jammers independently perform jamming on legitimate communication between transmitter and receiver. The main problem is finding the best channel for the SUs in order to avoid jammer's attacks on communication channels. We first designed a single game-based anti-jamming scheme that solves the problem of maximizing the long-term reward of the SU where communication channels are not used by the PUs and are not jammed by attackers. Then, we upgrade to a double game-based anti-jamming scheme in which the pre-selected channel is determined by using a single-game scheme. Afterward, through the pre-selected channel, the SU performs spectrum sensing to collect the PU status information, then, the second game will be solved using the updated accumulated reward. In addition, we adopted the transfer learning technique into the double-game scheme to accelerate the learning speed and improve network performance by exploiting the information learned in the double-game period. The simulation results validated that the proposed scheme can efficiently improve the long-term performance of the network. Through the proposed schemes, the optimal channel will be provided for SU to avoid jamming from attackers and significantly improve the security level of the CRN.

Thirdly, we investigate the transmit power optimization problem of SISO-SWIPT system with multi-user. The main goal is to minimize the sum of the transmit power of the transmitters by optimizing the PS ratios and the transmit power subjects to the constraints on required SINR and harvested energy. Since, the proposed transmit power optimization problem is a convex optimization problem, and thus, can be efficiently solved by Matlab's CVX. Afterward, a DL-based approach was suggested to optimal solution which obtained by optimization algorithm. In this work, optimal solutions can be approximated by DL architectures such as the DNN and RNNs. The simulation results showed that DL models can forecast output of the optimization problem effectively without prior knowledge about the system's state. Most of all, the DL-based approach provided low computation time, compared to the traditional optimization algorithm and is a potential solution for real-time resource allocation processing applications.

Finally, we consider the secure transmission optimization problem in IRS-assisted SWIPT system where the transmitter transmit a signal to the UE while an Eave' tries to listen information among them. By integrating an IRS to the system, eavesdropping can be reduced, and thus, network security can be enhanced. Moreover, the UE is equipped with a PS scheme where PS factor is an important factor that can prolong the operation time and improve the energy efficiency of devices. In this work, we aim to maximize the system secrecy rate by finding the optimal solutions for the transmission power, PS factor of the UE, and the phase shifts matrix of the IRS. We propose an AO-based scheme for solving the optimization problem where FPP, SCA, and penalty methods are used to solve the optimization problem. In addition, the computational efficiency of the proposed AO algorithm is also studied in comparison with the proposed DL-based approach. Specifically, 5 types of data and DNN structures are proposed. The simulation results verified that the system security rate of the system improved significantly with the help of IRS compared to the scheme without an IRS. In addition, the DL-based approach not only provided performance similar to that of the optimization algorithm but significantly improved computation time.

6.3 Future Directions

For future research directions regarding to AI-based techniques for enhancing wireless network performance, we consider several aspects as follows:

Regarding DL technique, although it helps to significantly improve the computational performance with a performance that approximates the optimization algorithm, DL will be done after when the optimal solutions are obtained through the optimization algorithm. After we get the trained network, we can get solutions in a real-time fashion, which will be sub-optimal. Here it is noticeable that optimal data set can be obtained offline by the optimization algorithm. In the case of RL, agents must get the information under a trial-and-error process to find an action in each state, because in the beginning, they have no prior information on the environment. As a result, the procedure could take a considerable amount of time for learning to reach an optimal policy. Thus, this makes it inefficient and inapplicable into large-scale networks where the state and action spaces will be larger. Recently, deep reinforcement learning (DRL) has emerged as an advanced version of RL, which is combined of RL and DNN. Specifically, DNNs can be used as an approximation function for mapping the system input (e.g., the system state) and the output in the RL task (e.g., the optimal policy). Consequently, DRL can overcome the limitations of RL, and thus provides better solutions to large-scale and sophisticated problems.

In CRN, experimental knowledge is needed to dispel the fictional part of CR. Therefore, putting "cognition" in the CRN is paramount at this moment. Fortunately, ML and AI will eventually become the brains of CR-supposedly any machine. Therefore, DRL can be applied to help address several aspects of CRN. For example in spectrum and channel management, decisions, such as which channel will be used and when it will be used, are made based on decision algorithms from the decision making module. In addition, the master-slave approach can be considered in CRN with GNU platform, in which the spectrum requesting messages is required to slave nodes by the master node. After that, the spectrum sensing results will be sent back to the master node by the slave nodes. The master node employs ML algorithms to choose the best band.

As we all know, the quality and availability of training datasets greatly affect the performance of some AI/ML algorithms, especially supervised learning and classification algorithms. There is necessary to investigate the impact of accuracy and data integrity on the performance gain of the AI/ML algorithms. In this way, we can compare the sensitivity of different AI/ML algorithms to inaccurate/incomplete data and then adopt the most appropriate algorithms according to the quality and availability of training datasets. Therefore, it is critically important to standardize the test cases, datasets, parameter sets and the corresponding interfaces and protocols to promote the deployment of AI in future wireless networks.

Next, when applying AI to wireless communications, an important issue to consider is the long latency caused by the slow convergence rate of AI algorithms. Therefore, latency control of various AI/ML algorithms needs to be investigated carefully, especially for those mission-critical applications such as remote surgery. To solve this problem, broad learning emerged as a complementary or alternative approach to the well-known DL algorithms. Broad learning system is designed based on the concept of random vector functional-link neural network (RVFL-NN). Because of the simple architecture, the long process of commonly employed deep architecture training can be eliminated. As an improvement of RVFL-NN, the input of broad learning system includes feature nodes created from the mapped features and enhancement nodes created from the enhanced features.

Although there is no doubt that AI is a promising technology for deployment in future communication networks, there are still some problems with large-scale deployment, especially, the security and reliability issue. This comes from the fact that DNNs are known to be vulnerable to adversaries and several recent studies have demonstrated the vulnerability of DRLs to such attacks. There are a few examples that show some of the malicious vulnerabilities when implementing AI in communication systems such as: what happens when a node attacks an AI-empowered device? What if some AI algorithms don't converge in some unexpected situations (e.g., abnormal traffic pattern or non-stationary channel conditions)? What is the potential risk when users share usage data with entities or network operator that implement AI algorithms? Therefore, for full confidence when applying AI-based methods in mission-critical real-world applications, understanding the vulnerabilities of these methods and address them is of primary concern in the deployment of AI-empowered systems.

Publications

International Journals

[1] H. T. Thien, H. Vu-Van, and I. Koo, “Implementation of Spectrum Sensing with Video Transmission for Cognitive Radio using USRP with GNU Radio,”*International Journal of Internet, Broadcasting and Communication*, vol. 10, no. 1, pp. 1–10, Feb. 2018.

[2] H. T. Thien, R. Tendeng, H. Vu-Van, and I. Koo, “Implementation of Spectrum-Sensing for Cognitive Radio Using USRP with GNU Radio and a Cloud Server,” *Journal of Information and Communication Convergence Engineering*, vol. 16, no. 1, pp. 23–30, Mar. 2018.

[3] H. T. Thien, V. -H. Vu, and I. Koo, “Game Theory-Based Smart Mobile-Data Offloading Scheme in 5G Cellular Networks,” *Applied Sciences*, vol. 10, no. 7, p. 2327, Mar. 2020.

[4] H. T. Thien, P. -V. Tuan, and I. Koo, “Deep Learning-Based Approach to Fast Power Allocation in SISO SWIPT Systems with a Power-Splitting Scheme,” *Applied Sciences*, vol. 10, no. 10, p. 3634, May 2020.

[5] H. T. Thien, V. -H. Vu, and I. Koo, “A Transfer Games Actor–Critic Learning Framework for Anti-Jamming in Multi-Channel Cognitive Radio Networks,” *IEEE Access*, vol. 9, pp. 47887–47900, 2021.

[6] H. T. Thien, P. -V. Tuan, and I. Koo, “A Secure-Transmission Maximization Scheme for SWIPT Systems Assisted by an Intelligent Reflecting Surface and Deep Learning,” *IEEE Access*, vol. 10, pp. 31851–31867, 2022.

[7] V. -H. Vu, H. T. Thien, and I. Koo, “A Repeated Games-Based Secure Multiple-

Channels Communications Scheme for Secondary Users with Randomly Attacking Eavesdroppers," *Applied Sciences*, vol. 9, no. 5, pp. 868, Feb. 2021.

Conferences

[8] H. T. Thien, P. -V. Tuan and I. Koo, "Deep Learning-based Secure Transmission for SWIPT System with Power-Splitting Scheme," *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, 2021, pp. 50–55, Jeju, Korea.

Bibliography

[1] Y.-C. Liang, K.-C. Chen, G. Y. Li, and P. Mahonen, "Cognitive radio networking and communications: an overview," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 3386–3407, 2011.

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

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

[4] X. Zhou, R. Zhang, and C. K. Ho, "Wireless information and power transfer: Architecture design and rate-energy tradeoff," *IEEE Transactions on Communications*, vol. 61, no. 11, pp. 4754–4767, 2013.

[5] I. Krikidis, S. Timotheou, S. Nikolaou, G. Zheng, D. W. K. Ng, and R. Schober, "Simultaneous wireless information and power transfer in modern communication systems," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 104–110, 2014.

[6] M. A. Abdulsattar and Z. A. Hussein, "Energy detection technique for spectrum sensing in cognitive radio: a survey," *International Journal of Computer Networks & Communications*, vol. 4, no. 5, pp. 223–242, 2012.

[7] J. Ghosh and I. Koo, "An approach to maximize throughput for energy efficient cognitive radio networks," *International Journal of Advanced Culture Technology*, vol. 1, no. 2, pp. 18–23, 2013.

[8] J. Ma, G. Y. Li, and B. H. Juang, "Signal processing in cognitive radio," *Proceedings of the IEEE*, vol. 97, no. 5, pp. 805–823, 2009.

[9] M. B. Weiss, "Spatio-temporal spectrum holes and the secondary user," *2011 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 216–222, 2011.

[10] J. C. Na, "Optimization in cooperative spectrum sensing," *Asia-pacific Journal of Convergent Research Interchange*, vol. 3, no. 1, pp. 19–31, 2017.

[11] D. Cabric, S. Mishra, and R. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*, vol. 1, 2004, pp. 772–776 Vol.1.

[12] J. H. Lee, H. Y. Kong *et al.*, "Primary user detection using a generalized selection combining over rayleigh fading channel," *International Journal of Internet, Broadcasting and Communication*, vol. 2, no. 2, pp. 1–3, 2010.

[13] S. Nimmi, V. Saranya, Theerthadas, and R. Gandhiraj, "Real-time video streaming using gstreamer in GNU radio platform," in *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)*, 2014, pp. 1–6.

[14] A. Khattab, D. Perkins, and M. Bayoumi, *Cognitive radio networks: from theory to practice.* Springer Science & Business Media, 2012.

[15] J.-P. Lang, "GNU radio–the free and open software radio ecosystem," *GNU Radio*, 2013.

[16] M. Sowmiya and M. Sangeetha, "Energy detection using NI USRP 2920," *International Journal of Science and Research*, vol. 5, no. 5, pp. 597–603, 2006.

[17] R. A. Rashid, M. A. Sarijari, N. Fisal, S. Yusof, N. H. Mahalin, and A. Lo, "Spectrum sensing measurement using GNU radio and USRP software radio platform," in *Proc. of The Seventh International Conference on Wireless and Mobile Communications*, 2011, pp. 237–242.

[18] S. Kuyoro, F. Osisanwo, and O. Akinsowon, "Internet of things (IoT): an overview," in *Proc. of the 3th International Conference on Advances in Engineering Sciences and Applied Mathematics (ICAESAM)*, 2015, pp. 23–24.

[19] M. A. G. Maureira, D. Oldenhof, and L. Teernstra, "Thingspeak–an API and web service for the internet of things," *World Wide Web*, 2011.

[20] H. V. Poor, *An introduction to signal detection and estimation.* Springer Science & Business Media, 2013.

[21] T. A. Tuan and I. Koo, "An efficient channel selection algorithm for cognitive radio sensor networks," *International Journal of Internet, Broadcasting and Communication*, vol. 3, no. 2, pp. 26–30, 2010.

[22] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer networks*, vol. 50, no. 13, pp. 2127–2159, 2006.

[23] B. Mumey, J. Tang, I. R. Judson, and D. Stevens, "On routing and channel selection in cognitive radio mesh networks," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 9, pp. 4118–4128, 2012.

[24] A. Jamal, C.-K. Tham, and W.-C. Wong, "Dynamic packet size optimization and channel selection for cognitive radio sensor networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 1, no. 4, pp. 394–405, 2015.

[25] M. Ju and K.-M. Kang, "Cognitive radio networks with secondary network selection," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pp. 966–972, 2016.

[26] M. Xu, M. Jin, Q. Guo, and Y. Li, "Multichannel selection for cognitive radio networks with rf energy harvesting," *IEEE Wireless Communications Letters*, vol. 7, no. 2, pp. 178–181, 2018.

[27] A. Sabbah, M. Ibnkahla, O. Issa, and B. Doray, "Control channel selection techniques in cognitive radio networks: A comparative performance analysis," *Journal of Communications and Networks*, vol. 20, no. 1, pp. 57–68, 2018.

[28] J. Ren, H. Zhang, X. Liu, and Y. Qin, "Energy efficiency-centric channel selecting in energy harvesting cognitive radio sensor network," in *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 1, 2019, pp. 2736–2739.

[29] P. Alcoy, S. Bjarnason, P. Bowen, C. Chui, K. Kasavchenko, and G. Sockrider, "13th annual worldwide infrastructure security report," *NETSCOUT Arbor*, 2018.

[30] J. Li, Z. Feng, Z. Feng, and P. Zhang, "A survey of security issues in cognitive radio networks," *China Communications*, vol. 12, no. 3, pp. 132–150, 2015.

[31] A. G. Fragkiadakis, E. Z. Tragos, and I. G. Askoxylakis, "A survey on security threats and detection techniques in cognitive radio networks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 428–445, 2013.

[32] Z. Bai, L. Ma, Y. Dong, P. Ma, and Y. Ma, "Energy-efficient resource allocation for secure cognitive radio network with delay QoS guarantee," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2795–2805, 2019.

[33] D. T. Hoang, D. Niyato, P. Wang, and D. I. Kim, "Performance analysis of wireless energy harvesting cognitive radio networks under smart jamming attacks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 1, no. 2, pp. 200–216, 2015.

[34] Y. Wu, B. Wang, K. J. R. Liu, and T. C. Clancy, "Anti-jamming games in multi-channel cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 1, pp. 4–15, 2012.

[35] L. Xiao, J. Liu, Q. Li, N. B. Mandayam, and H. V. Poor, "User-centric view of jamming games in cognitive radio networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2578–2590, 2015.

[36] B. Wang, Y. Wu, and K. R. Liu, "Game theory for cognitive radio networks: An overview," *Computer networks*, vol. 54, no. 14, pp. 2537–2561, 2010.

[37] A. Neyman, S. Sorin, and S. Sorin, *Stochastic games and applications.* Springer Science & Business Media, 2003, vol. 570.

[38] J. Filar and K. Vrieze, *Competitive Markov decision processes.* Springer Science & Business Media, 2012.

[39] X. Xie and W. Wang, "Detecting primary user emulation attacks in cognitive radio networks via physical layer network coding," *Procedia Computer Science*, vol. 21, pp. 430–435, 2013.

[40] V.-H. Vu, H. T. Thien, and I. Koo, "A repeated games-based secure multiple-channels communications scheme for secondary users with randomly attacking eavesdroppers," *Applied Sciences*, vol. 9, no. 5, p. 868, 2019.

[41] 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, 2016.

[42] 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, 2012.

[43] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[44] B. Wang, Y. Wu, K. R. Liu, and T. C. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 877–889, 2011.

[45] S. Singh and A. Trivedi, "Anti-jamming in cognitive radio networks using reinforcement learning algorithms," in *2012 Ninth International Conference on Wireless and Optical Communications Networks (WOCN)*, 2012, pp. 1–5.

[46] G. Han, L. Xiao, and H. V. Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2087–2091.

[47] Y. Bi, Y. Wu, and C. Hua, "Deep reinforcement learning based multi-user anti-jamming strategy," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[48] J. Xu, H. Lou, W. Zhang, and G. Sang, "An intelligent anti-jamming scheme for cognitive radio based on deep reinforcement learning," *IEEE Access*, vol. 8, pp. 202 563–202 572, 2020.

[49] T. Takano, H. Takase, H. Kawanaka, H. Kita, T. Hayashi, and S. Tsuruoka, "Transfer learning based on forbidden rule set in actor-critic method," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 5, pp. 2907–2917, 2011.

[50] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey." *Journal of Machine Learning Research*, vol. 10, no. 7, 2009.

[51] D. W. Aha, M. Molineaux, and G. Sukthankar, "Case-based reasoning in transfer learning," in *International Conference on Case-Based Reasoning.* Springer, 2009, pp. 29–44.

[52] 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, 2010.

[53] L. A. C. Jr., J. P. Matsuura, R. L. de Màntaras, and R. A. Bianchi, "Using transfer learning to speed-up reinforcement learning: A cased-based approach," in *2010 Latin American Robotics Symposium and Intelligent Robotics Meeting*, 2010, pp. 55–60.

[54] 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, 2014.

[55] K. A.M., F. Hu, and S. Kumar, "Intelligent spectrum management based on transfer actor-critic learning for rateless transmissions in cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 5, pp. 1204–1215, 2018.

[56] G. Chen, Y. Zhan, Y. Chen, L. Xiao, Y. Wang, and N. An, "Reinforcement learning based power control for in-body sensors in wbans against jamming," *IEEE Access*, vol. 6, pp. 37 403–37 412, 2018.

[57] C. Dai, L. Xiao, X. Wan, and Y. Chen, "Reinforcement learning with safe exploration for network security," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3057–3061.

[58] X. Lu, L. Xiao, C. Dai, and H. Dai, "UAV-aided cellular communications with deep reinforcement learning against jamming," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 48–53, 2020.

[59] W. Zhang, R. K. Mallik, and K. B. Letaief, "Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 5761–5766, 2009.

[60] Z. Quan, S. Cui, and A. H. Sayed, "Optimal linear cooperation for spectrum sensing in cognitive radio networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 28–40, 2008.

[61] M. G. Oskoui, P. Khorramshahi, and J. A. Salehi, "Using game theory to battle jammer in control channels of cognitive radio ad hoc networks," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–5.

[62] M. T. Goodrich and R. Tamassia, *Algorithm design and applications.* Wiley Hoboken, 2015.

[63] M. Hajimirsaadeghi and N. B. Mandayam, "A dynamic colonel blotto game model for spectrum sharing in wireless networks," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2017, pp. 287–294.

[64] C. Jiang, Y. Chen, K. J. R. Liu, and Y. Ren, "Optimal pricing strategy for operators in cognitive femtocell networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 9, pp. 5288–5301, 2014.

[65] Y. Zhu, D. Zhao, and X. Li, "Iterative adaptive dynamic programming for solving unknown nonlinear zero-sum game based on online data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 714–725, 2017.

[66] J. V. Stone, *Bayes' Rule: A Tutorial Introduction to Bayesian Analysis.* Sebtel Press, 2013.

[67] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[68] Y. Xu, C. Shen, Z. Ding, X. Sun, S. Yan, G. Zhu, and Z. Zhong, "Joint beamforming and power-splitting control in downlink cooperative SWIPT NOMA systems," *IEEE Transactions on Signal Processing*, vol. 65, no. 18, pp. 4874–4886, 2017.

[69] X. Lu, P. Wang, D. Niyato, D. I. Kim, and Z. Han, “Wireless charging technologies: Fundamentals, standards, and network applications,” *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1413–1452, 2016.

[70] L. R. Varshney, “Transporting information and energy simultaneously,” in *2008 IEEE International Symposium on Information Theory*, 2008, pp. 1612–1616.

[71] R. Zhang and C. K. Ho, “MIMO broadcasting for simultaneous wireless information and power transfer,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 1989–2001, 2013.

[72] L. Liu, R. Zhang, and K.-C. Chua, “Wireless information and power transfer: A dynamic power splitting approach,” *IEEE Transactions on Communications*, vol. 61, no. 9, pp. 3990–4001, 2013.

[73] Q. Shi, C. Peng, W. Xu, and Y. Wang, “Joint transceiver design for MISO SWIPT interference channel,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4753–4757.

[74] C. Shen, W.-C. Li, and T.-H. Chang, “Wireless information and energy transfer in multi-antenna interference channel,” *IEEE Transactions on Signal Processing*, vol. 62, no. 23, pp. 6249–6264, 2014.

[75] M. R. A. Khandaker and K.-K. Wong, “SWIPT in MISO multicasting systems,” *IEEE Wireless Communications Letters*, vol. 3, no. 3, pp. 277–280, 2014.

[76] Q. Shi, L. Liu, W. Xu, and R. Zhang, “Joint transmit beamforming and receive power splitting for MISO SWIPT systems,” *IEEE Transactions on Wireless Communications*, vol. 13, no. 6, pp. 3269–3280, 2014.

[77] J. Park and B. Clerckx, “Joint wireless information and energy transfer in a two-user MIMO interference channel,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 8, pp. 4210–4221, 2013.

[78] ——, “Joint wireless information and energy transfer in a K-user MIMO interference channel,” *IEEE Transactions on Wireless Communications*, vol. 13, no. 10, pp. 5781–5796, 2014.

[79] ——, “Joint wireless information and energy transfer with reduced feedback in MIMO interference channels,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 8, pp. 1563–1577, 2015.

[80] S. Lee, L. Liu, and R. Zhang, “Collaborative wireless energy and information transfer in interference channel,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 545–557, 2015.

[81] P.-V. Tuan and I. Koo, “Efficient transceiver design for large-scale SWIPT system with time-switching and power-splitting receivers,” *IEICE Transactions on Communications*, vol. 101, no. 7, pp. 1744–1751, 2018.

[82] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, “Learning to optimize: Training deep neural networks for wireless resource management,” in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2017, pp. 1–6.

[83] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, “An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel,” *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[84] H. Baligh, M. Hong, W.-C. Liao, Z.-Q. Luo, M. Razaviyayn, M. Sanjabi, and R. Sun, “Cross-layer provision of future cellular networks: A WMMSE-based approach,” *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 56–68, 2014.

[85] W. Yu, G. Ginis, and J. Cioffi, “Distributed multiuser power control for digital subscriber lines,” *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 5, pp. 1105–1115, 2002.

[86] D. A. Schmidt, C. Shi, R. A. Berry, M. L. Honig, and W. Utschick, “Distributed resource allocation schemes,” *IEEE Signal Processing Magazine*, vol. 26, no. 5, pp. 53–63, 2009.

[87] Z.-q. Luo, W.-k. Ma, A. M.-c. So, Y. Ye, and S. Zhang, “Semidefinite relaxation of quadratic optimization problems,” *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.

[88] N. Sidiropoulos, T. Davidson, and Z.-Q. Luo, “Transmit beamforming for physical-layer multicasting,” *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2239–2251, 2006.

[89] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[90] T. J. O’Shea, J. Corgan, and T. C. Clancy, “Convolutional radio modulation recognition networks,” in *International conference on engineering applications of neural networks*. Springer, 2016, pp. 213–226.

[91] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[92] H. Ye, G. Y. Li, and B.-H. Juang, “Power of deep learning for channel estimation and signal detection in OFDM systems,” *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.

[93] X. Wang, L. Gao, S. Mao, and S. Pandey, “CSI-based fingerprinting for indoor localization: A deep learning approach,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763–776, 2017.

[94] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[95] M. Kim, W. Lee, and D.-H. Cho, “A novel papr reduction scheme for OFDM system based on deep learning,” *IEEE Communications Letters*, vol. 22, no. 3, pp. 510–513, 2018.

[96] S. Dörner, S. Cammerer, J. Hoydis, and S. t. Brink, “Deep learning based communication over the air,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2018.

[97] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[98] P. Sharma, A. Raghuvanshi, and R. Pachori, *Artificial intelligence and soft computing: Soft computing techniques: Artificial intelligence, neural networks, fuzzy logic and genetic algorithm.* Educreation Publishing, 2019.

[99] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[100] F. H. Nordin and F. H. Nagi, "Layer-recurrent network in identifying a nonlinear system," in *2008 International Conference on Control, Automation and Systems*, 2008, pp. 387–391.

[101] O. Castillo, P. Melin, W. Pedrycz, and J. Kacprzyk, *Recent Advances on Hybrid Approaches for Designing Intelligent Systems.* Springer, 2014.

[102] G. Carmignani, M. Passacantando, and G. Tumminelli, "A novel method based on artificial neural network to production planning: a case study of a paints producer," in *Proceedings of the Eighteenth International Working Seminar on Production Economics, Innsbruck, Austria*, vol. 28, 2014.

[103] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[104] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[105] S. Liu, G. Liao, and Y. Ding, "Stock transaction prediction modeling and analysis based on LSTM," in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2018, pp. 2787–2790.

[106] A. Sarah, K. Lee, and H. Kim, "LSTM model to forecast time series for ec2 cloud price," in *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, 2018, pp. 1085–1088.

[107] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” 2014.

[108] M. F. Møller, “A scaled conjugate gradient algorithm for fast supervised learning,” *Neural networks*, vol. 6, no. 4, pp. 525–533, 1993.

[109] H.-A. Ounifi, A. Gherbi, N. Kara, and W. Li, “A deep neural network based approach to energy efficiency analysis for cloud data center,” in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1, 2019, pp. 1397–1404.

[110] S. Troia, R. Alvizu, Y. Zhou, G. Maier, and A. Pattavina, “Deep learning-based traffic prediction for network optimization,” in *2018 20th International Conference on Transparent Optical Networks (ICTON)*, 2018, pp. 1–4.

[111] Y. Chen, Y. Shi, and B. Zhang, “Modeling and optimization of complex building energy systems with deep neural networks,” in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, 2017, pp. 1368–1373.

[112] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, “Learning to optimize: Training deep neural networks for interference management,” *IEEE Transactions on Signal Processing*, vol. 66, no. 20, pp. 5438–5453, 2018.

[113] G. Chen, J. Tang, and J. P. Coon, “Optimal routing for multihop social-based D2D communications in the internet of things,” *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1880–1889, 2018.

[114] Q. Wu and R. Zhang, “Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network,” *IEEE Communications Magazine*, vol. 58, no. 1, pp. 106–112, 2020.

[115] J. Zhao, “A survey of intelligent reflecting surfaces (IRSs): Towards 6G wireless communication networks,” *arXiv preprint arXiv:1907.04789*, 2019.

[116] S. Hu, F. Rusek, and O. Edfors, “Beyond massive MIMO: The potential of data transmission with large intelligent surfaces,” *IEEE Transactions on Signal Processing*, vol. 66, no. 10, pp. 2746–2758, 2018.

[117] ——, “Beyond massive MIMO: The potential of positioning with large intelligent surfaces,” *IEEE Transactions on Signal Processing*, vol. 66, no. 7, pp. 1761–1774, 2018.

[118] Z.-Q. He and X. Yuan, “Cascaded channel estimation for large intelligent metasurface assisted massive MIMO,” *IEEE Wireless Communications Letters*, vol. 9, no. 2, pp. 210–214, 2020.

[119] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, “Reconfigurable intelligent surfaces for energy efficiency in wireless communication,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 8, pp. 4157–4170, 2019.

[120] M. Di Renzo, A. Zappone, M. Debbah, M.-S. Alouini, C. Yuen, J. de Rosny, and S. Tretyakov, “Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 11, pp. 2450–2525, 2020.

[121] A. Mukherjee, S. A. A. Fakoorian, J. Huang, and A. L. Swindlehurst, “Principles of physical layer security in multiuser wireless networks: A survey,” *IEEE Communications Surveys Tutorials*, vol. 16, no. 3, pp. 1550–1573, 2014.

[122] S. Goel and R. Negi, “Guaranteeing secrecy using artificial noise,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, pp. 2180–2189, 2008.

[123] A. Khisti and G. W. Wornell, “Secure transmission with multiple antennas I: The MISOME wiretap channel,” *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3088–3104, 2010.

[124] L. Liu, R. Zhang, and K.-C. Chua, “Secrecy wireless information and power transfer with MISO beamforming,” *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1850–1863, 2014.

[125] J. Tang, T. Dai, M. Cui, X. Y. Zhang, A. Shojaeifard, K.-K. Wong, and Z. Li, “Optimization for maximizing sum secrecy rate in SWIPT-enabled NOMA systems,” *IEEE Access*, vol. 6, pp. 43 440–43 449, 2018.

[126] H. Shen, W. Xu, S. Gong, Z. He, and C. Zhao, “Secrecy rate maximization for intelligent reflecting surface assisted multi-antenna communications,” *IEEE Communications Letters*, vol. 23, no. 9, pp. 1488–1492, 2019.

[127] M. Cui, G. Zhang, and R. Zhang, "Secure wireless communication via intelligent reflecting surface," *IEEE Wireless Communications Letters*, vol. 8, no. 5, pp. 1410–1414, 2019.

[128] S. Zargari, S. Farahmand, and B. Abolhassani, "Joint design of transmit beamforming, IRS platform, and power splitting SWIPT receivers for downlink cellular multiuser MISO," *Physical Communication*, vol. 48, p. 101413, 2021.

[129] Z. Li, W. Chen, Q. Wu, K. Wang, and J. Li, "Joint beamforming design and power splitting optimization in IRS-assisted SWIPT NOMA networks," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2021.

[130] S. Zargari, S. Farahmand, B. Abolhassani, and C. Tellambura, "Robust active and passive beamformer design for IRS-aided downlink MISO PS-SWIPT with a nonlinear energy harvesting model," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 4, pp. 2027–2041, 2021.

[131] H. Niu, Z. Chu, F. Zhou, Z. Zhu, L. Zhen, and K.-K. Wong, "Robust design for intelligent reflecting surface assisted secrecy SWIPT network," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2021.

[132] J. Yang, X. Ji, K. Huang, X. Sun, and Y. Wang, "Enhanced secure SWIPT in heterogeneous network via intelligent reflecting surface," *Security and Communication Networks*, vol. 2021, 2021.

[133] Z. Deng and Y. Pan, "Optimal beamforming for IRS-assisted SWIPT system with an energy-harvesting eavesdropper," *Electronics*, vol. 10, no. 20, p. 2536, 2021.

[134] W. Sun, Q. Song, L. Guo, and J. Zhao, "Secrecy rate maximization for intelligent reflecting surface aided SWIPT systems," in *2020 IEEE/CIC International Conference on Communications in China (ICCC)*, 2020, pp. 1276–1281.

[135] O. Mehanna, K. Huang, B. Gopalakrishnan, A. Konar, and N. D. Sidiropoulos, "Feasible point pursuit and successive approximation of non-convex QCQPs," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 804–808, 2015.

[136] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, "Intelligent reflecting surface-aided wireless communications: A tutorial," *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3313–3351, 2021.

[137] Y. Liang, G. Kramer, and H. V. Poor, "Compound wiretap channels," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, pp. 1–12, 2009.

[138] Q. Li and W.-K. Ma, "Spatially selective artificial-noise aided transmit optimization for MISO multi-eves secrecy rate maximization," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2704–2717, 2013.

[139] 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, 2013.

[140] D. W. K. Ng, E. S. Lo, and R. Schober, "Robust beamforming for secure communication in systems with wireless information and power transfer," *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4599–4615, 2014.

[141] J. Wang and D. P. Palomar, "Worst-case robust MIMO transmission with imperfect channel knowledge," *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 3086–3100, 2009.

[142] D. Xu, X. Yu, Y. Sun, D. W. K. Ng, and R. Schober, "Resource allocation for IRS-assisted full-duplex cognitive radio systems," *IEEE Transactions on Communications*, vol. 68, no. 12, pp. 7376–7394, 2020.

[143] M. Bloch, J. Barros, M. R. D. Rodrigues, and S. W. McLaughlin, "Wireless information-theoretic security," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2515–2534, 2008.

[144] J. Chen, Y.-C. Liang, H. V. Cheng, and W. Yu, "Channel estimation for reconfigurable intelligent surface aided multi-user MIMO systems," *arXiv preprint arXiv:1912.03619*, 2019.

[145] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5394–5409, 2019.

[146] D. Tse and P. Viswanath, *Fundamentals of wireless communication.* Cambridge university press, 2005.

[147] T. Nakanishi and M. Kitano, “Storage and retrieval of electromagnetic waves using electromagnetically induced transparency in a nonlinear metamaterial,” *Applied Physics Letters*, vol. 112, no. 20, p. 201905, 2018.

[148] S. Zhang and R. Zhang, “Capacity characterization for intelligent reflecting surface aided MIMO communication,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1823–1838, 2020.

[149] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, March 2004.

[150] I. Pólik and T. Terlaky, “Interior point methods for nonlinear optimization,” in *Nonlinear optimization.* Springer, 2010, pp. 215–276.

[151] Y. Sun, P. Babu, and D. P. Palomar, “Majorization-minimization algorithms in signal processing, communications, and machine learning,” *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 794–816, 2017.

[152] S. Mao, S. Leng, J. Hu, and K. Yang, “Power minimization resource allocation for underlay MISO-NOMA SWIPT systems,” *IEEE Access*, vol. 7, pp. 17 247–17 255, 2019.

[153] P. Viet Tuan, P. Ngoc Son, T. Trung Duy, S. Q. Nguyen, V. Q. B. Ngo, V. Do Quang, I. Koo *et al.*, “Optimizing a secure two-way network with non-linear SWIPT, channel uncertainty, and a hidden eavesdropper,” *Electronics*, vol. 9, no. 8, p. 1222, 2020.

[154] A. H. Phan, H. D. Tuan, H. H. Kha, and D. T. Ngo, “Nonsmooth optimization for efficient beamforming in cognitive radio multicast transmission,” *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2941–2951, 2012.

[155] D. Munoz, F. B. Lara, C. Vargas, and R. Enriquez-Caldera, *Position location techniques and applications.* Academic Press, 2009.